EE 604, Digital Image Processing

Chapter 2: Digital Image Fundamentals

Dr. W. David Pan
Dept. of ECE
UAH

Topics

- Human visual system; Image formation in the eye and its capabilities for brightness adaptation and discrimination.
- Light, other components of the electromagnetic spectrum, and their imaging characteristics
- Imaging sensors and how they are used to generate digital images
- The concepts of uniform image sampling and intensity quantization
- Digital image representation, the effects of varying the number of samples and intensity levels in an image
- The concepts of spatial and intensity resolution
- The principles of image interpolation
- Basic relationships between pixels
- Principal mathematical tools
- Basics of Matlab IPT

Visual Perception

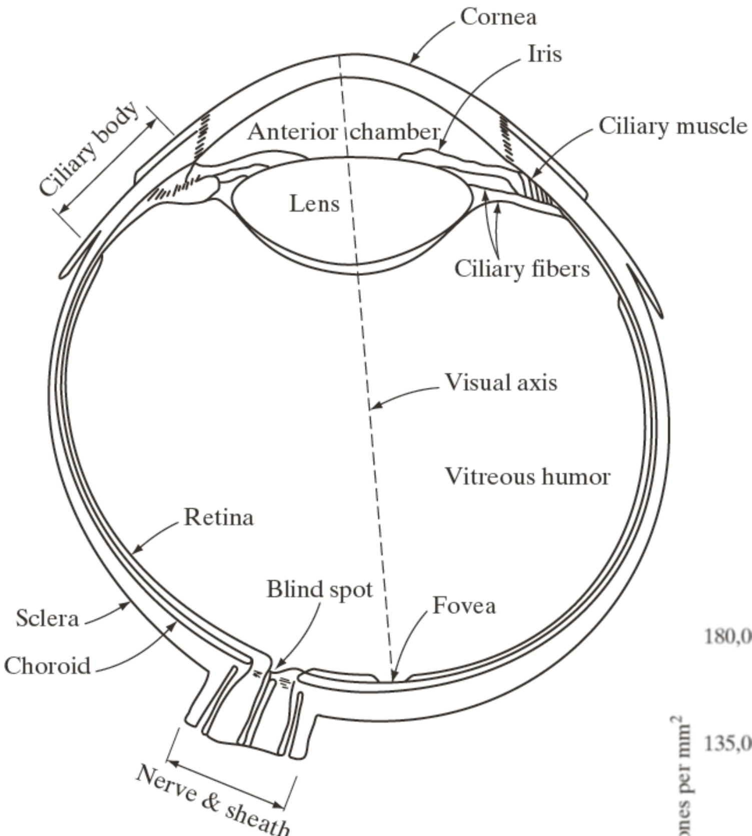


FIGURE 2.1 Simplified diagram of a cross section of the human eye.

Two types of light receptors

- Cones
  - Highly sensitive to color
  - Cone vision is called photopic or bright-light vision
- Rods
  - Gives a general, overall picture of the field of view
  - Rod vision is called scotopic or dim-light vision

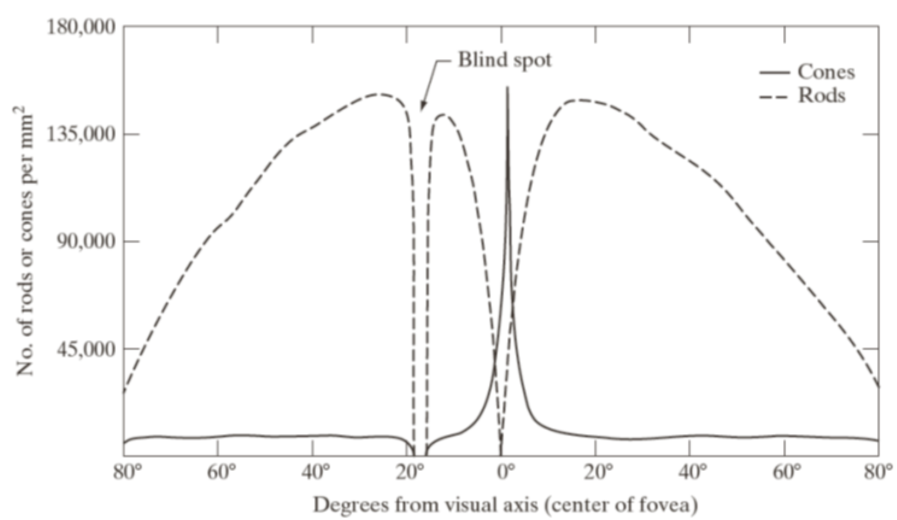


FIGURE 2.2 Distribution of rods and cones in the retina.

Image Formation in the Eye

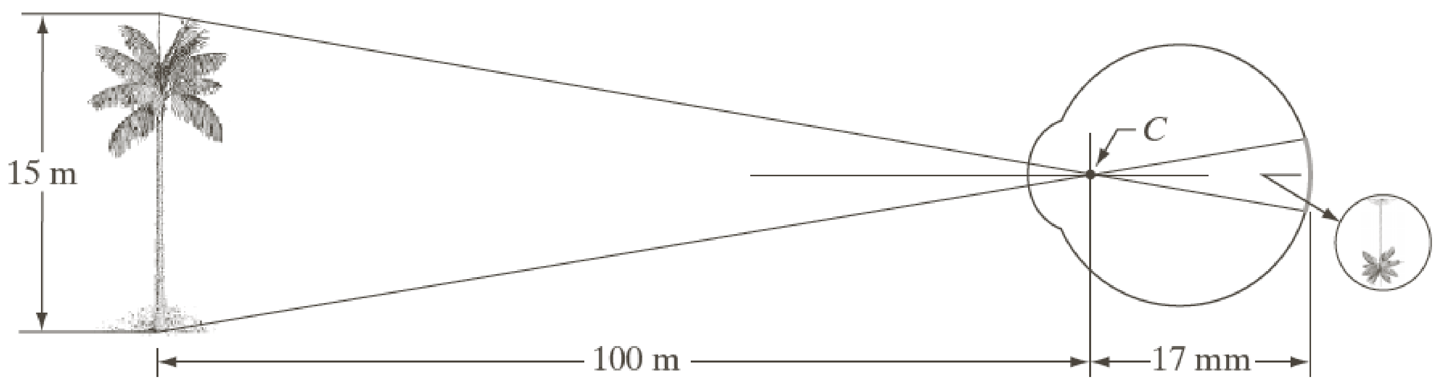


FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens.

Brightness Adaptation

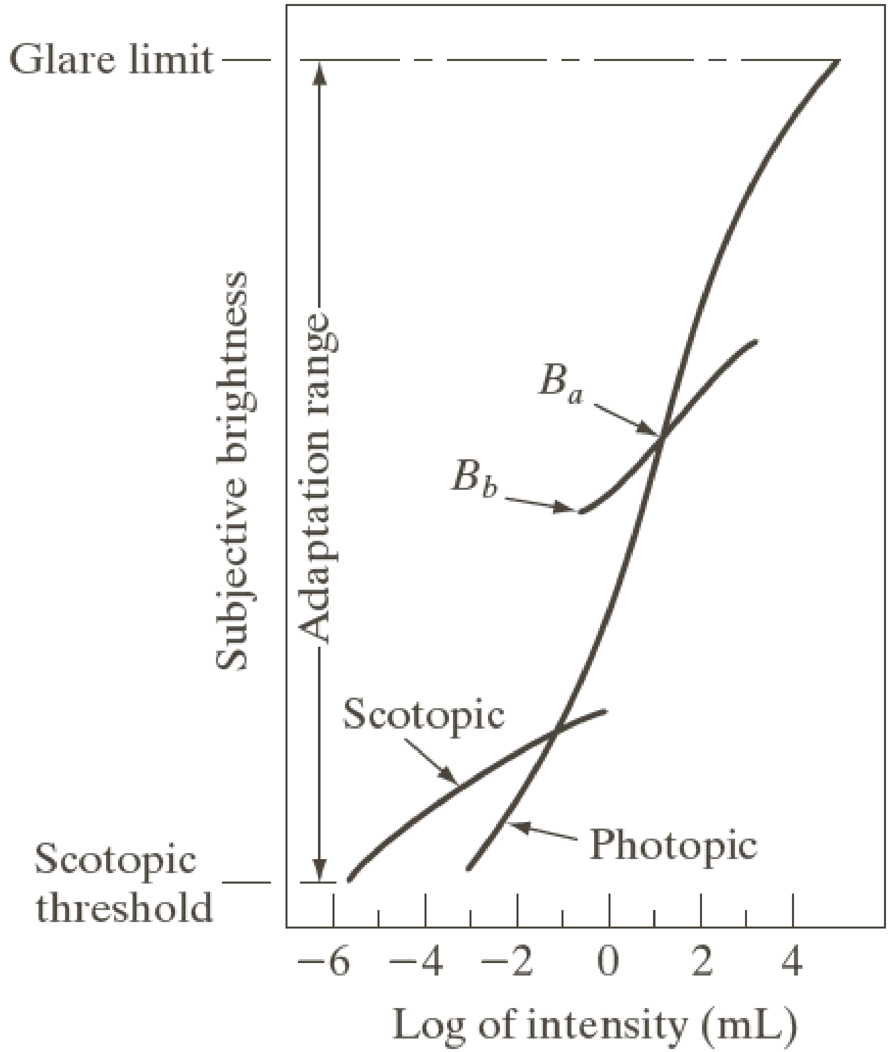


FIGURE 2.4 Range of subjective brightness sensations showing a particular adaptation level.

Brightness Discrimination

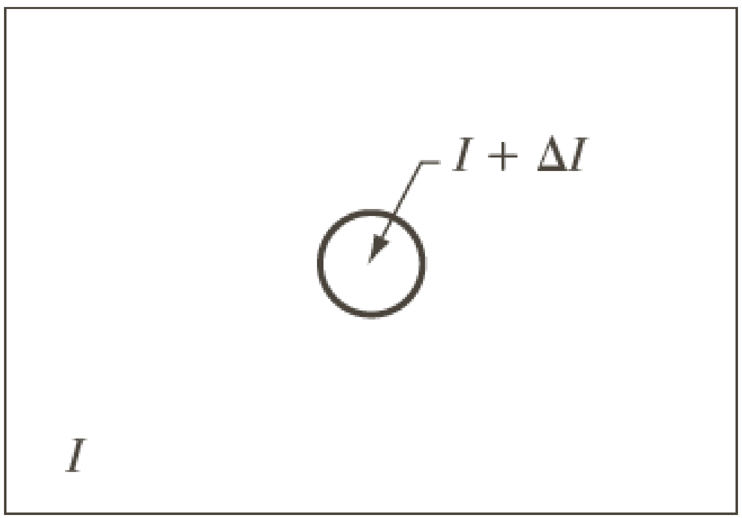


FIGURE 2.5 Basic experimental setup used to characterize brightness discrimination.

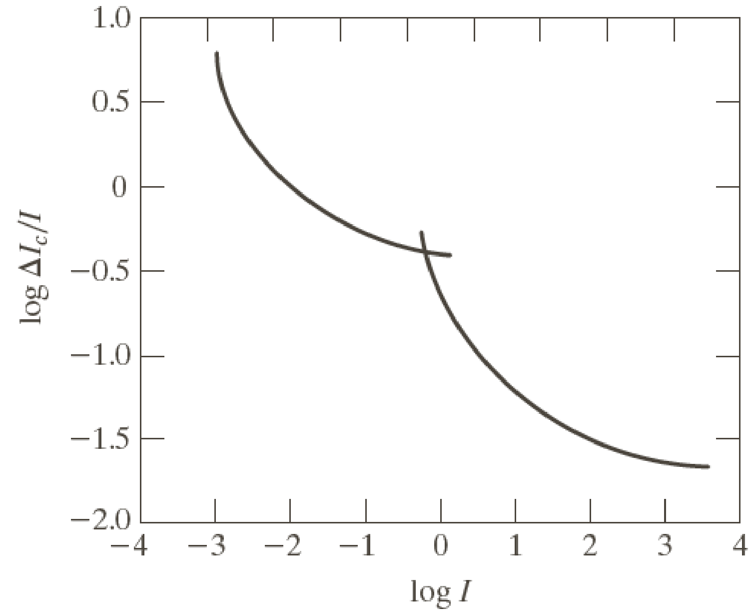


FIGURE 2.6 Typical Weber ratio as a function of intensity.

Brightness is not a simple function of intensity.

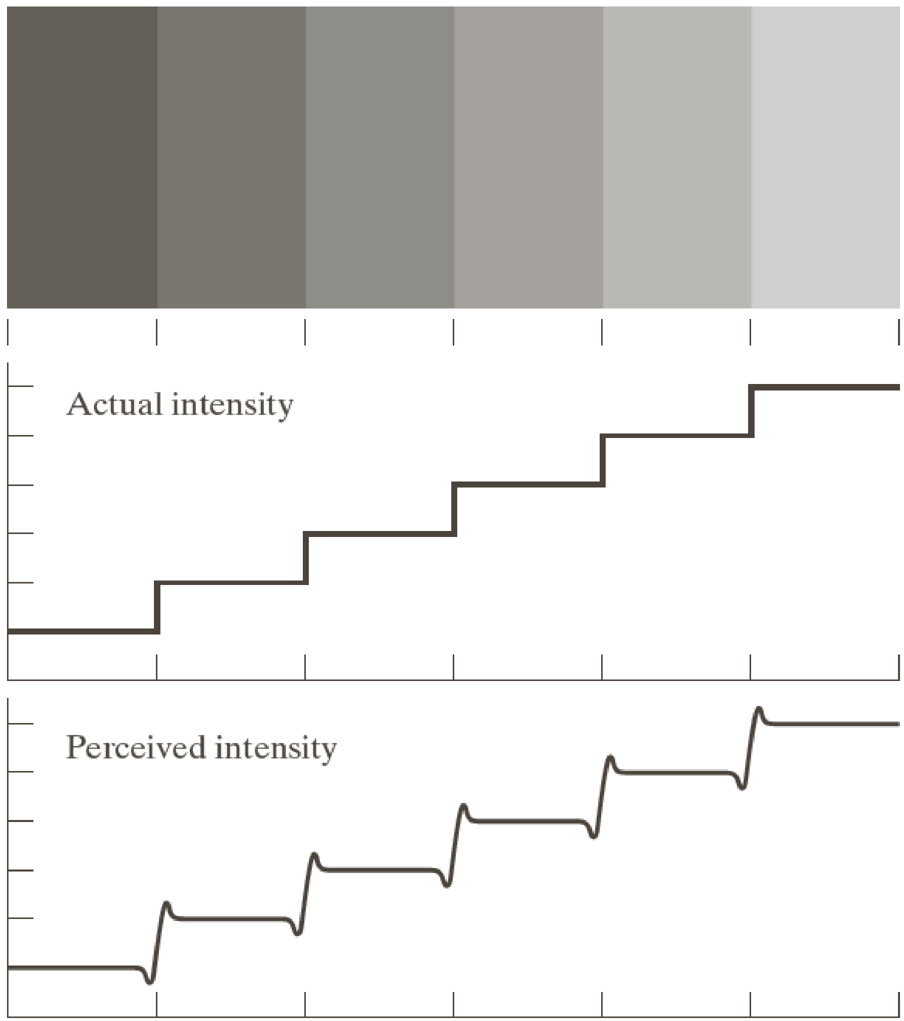
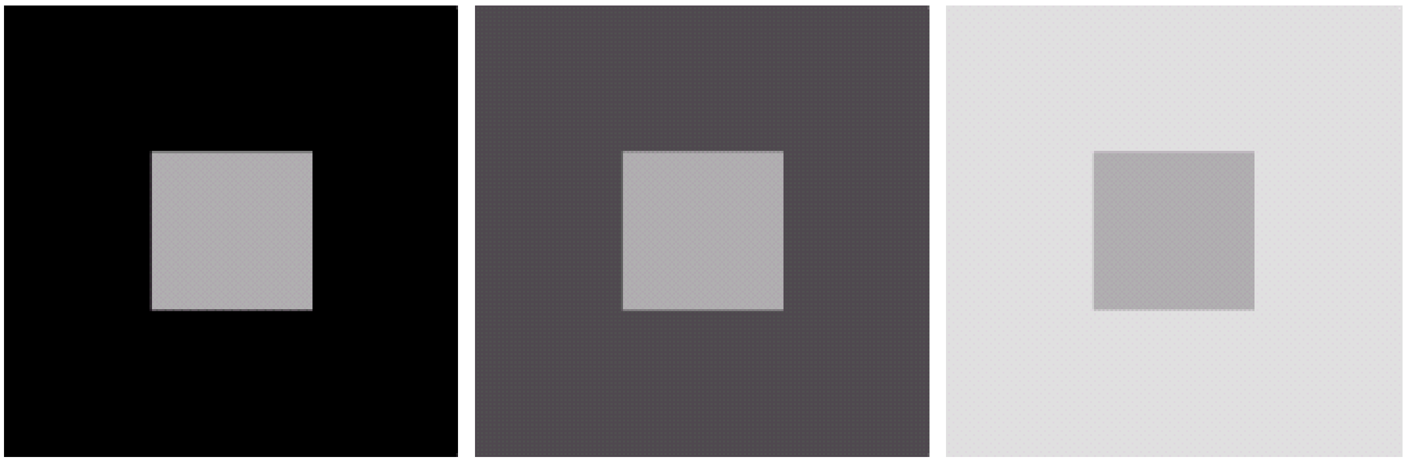


FIGURE 2.7 Illustration of the Mach band effect. Perceived intensity is not a simple function of actual intensity.

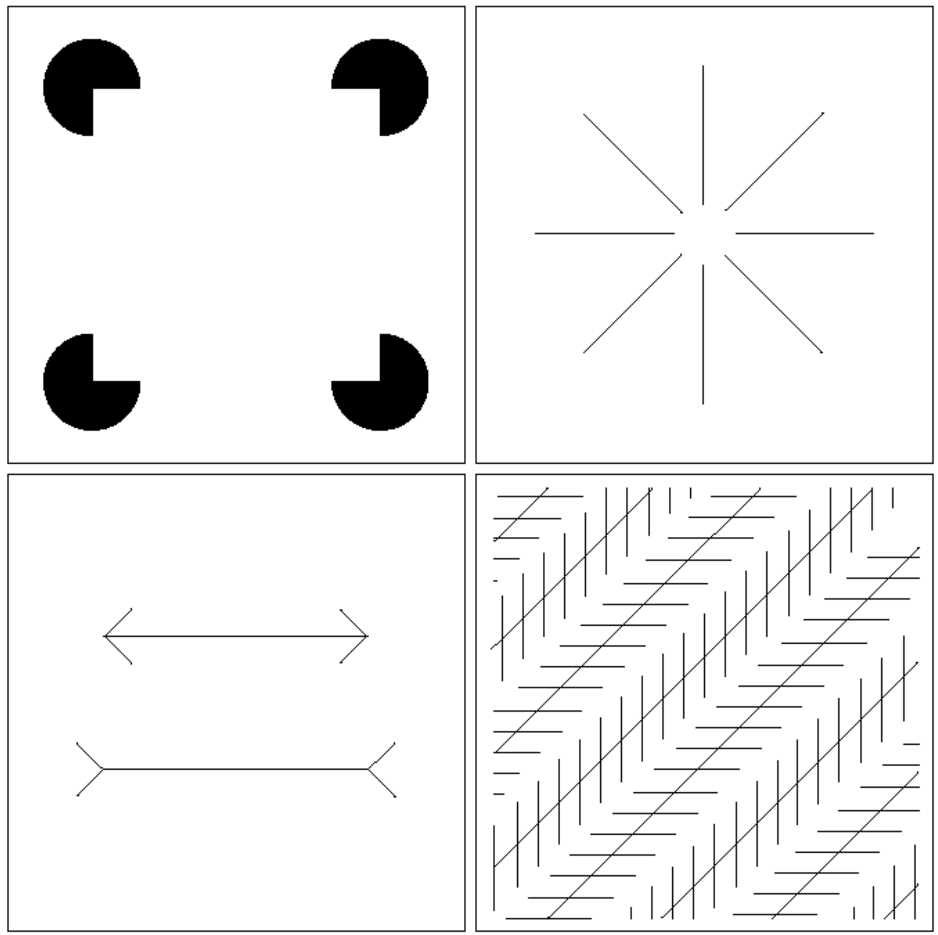
Simultaneous Contrast



a b c

FIGURE 2.8 Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

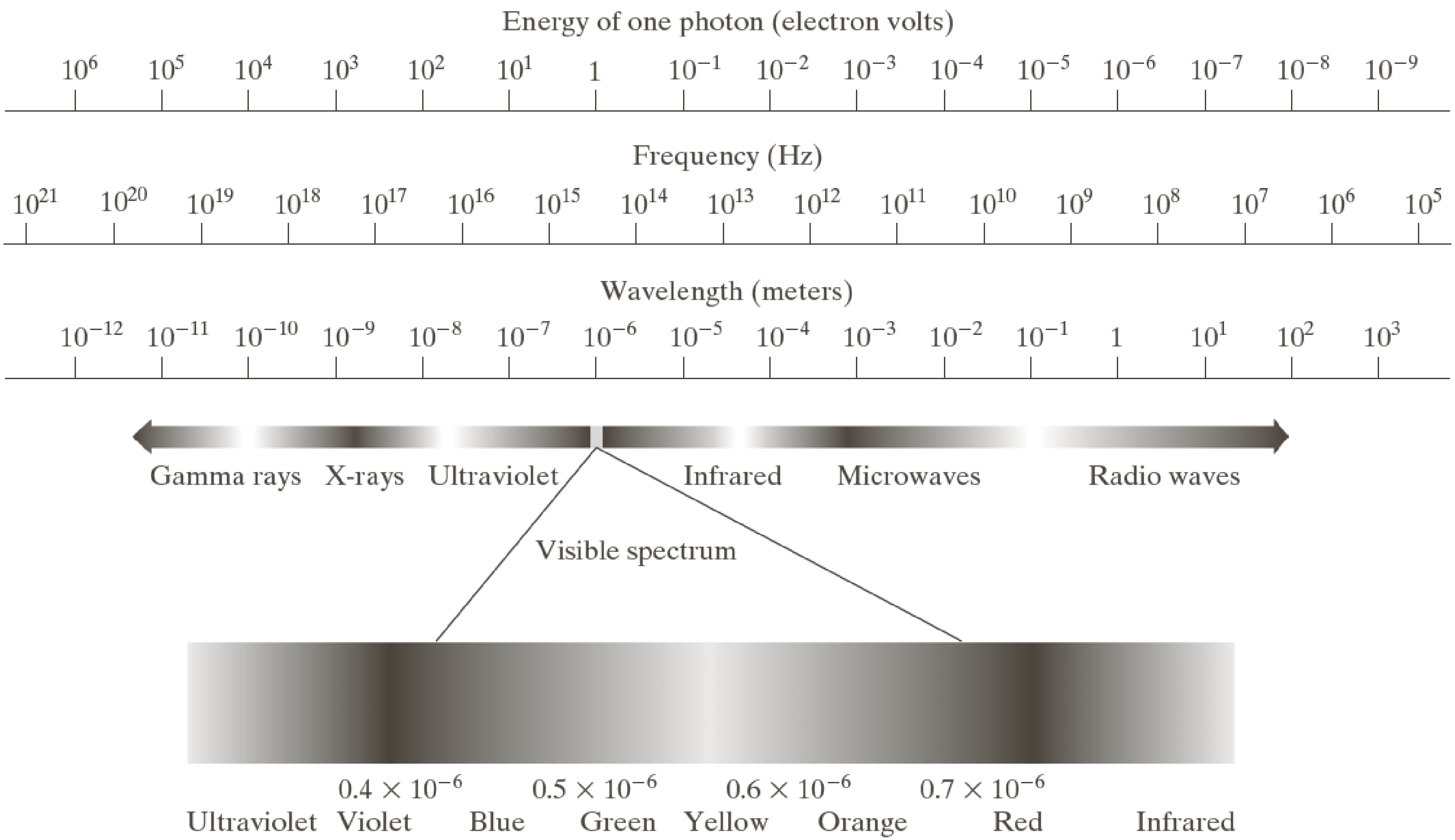
Optical Illusions



a b
c d

FIGURE 2.9 Some well-known optical illusions.

Light and Electromagnetic Spectrum

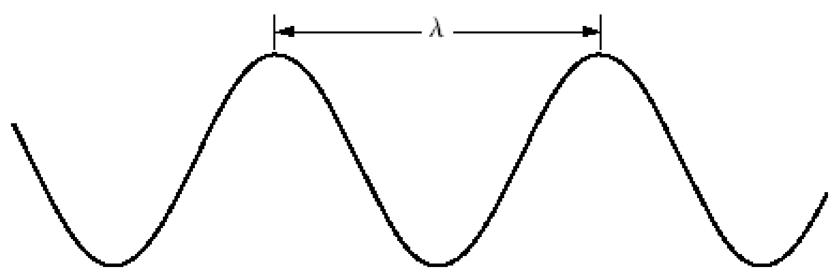


Wavelength(λ), Frequency(ν), Energy(E)

$$\lambda = \frac{c}{\nu} \qquad c = 2.988 \times 10^8 \ m/s$$

$$E = h\nu \qquad h = 6.626 \times 10^{-34} \ J \cdot s$$

FIGURE 2.11
Graphical representation of one wavelength.



Color

- The colors that humans perceive in an object are determined by the nature of the light *reflected* from the object.
  - A body that reflects light relatively balanced in all visible wavelengths appears white to the observer.
  - However, a body that favors reflectance in a limited range of the visible spectrum exhibits some shades of color.
  - For example, green objects reflect light with wavelengths primarily in the 500 to 570 nm range while absorbing most of the energy at other wavelengths.

Gray-Scale Images

- Light that is void of color is called **monochromatic** (or achromatic) light. The only attribute of monochromatic light is its intensity or amount.
- Because the intensity of monochromatic light is perceived to vary from black to grays and finally to white, the term **gray level** is used commonly to denote monochromatic intensity. We use the terms **intensity** and **gray level** interchangeably in subsequent discussions.
- The range of measured values of monochromatic light from black to white is usually called the gray scale, and monochromatic images are frequently referred to as **gray-scale** images.

Chromatic Light Source

- *Chromatic (color) light* spans the electromagnetic energy spectrum from approximately 0.43 to 0.79 μm.
- In addition to frequency, three basic quantities are used to describe the quality of a chromatic light source:
- ***Radiance*** is the total amount of energy that flows from the light source, and it is usually measured in watts (W).
- ***Luminance,*** measured in lumens (lm), gives a measure of the amount of energy an observer *perceives* from a light source.
  - For example, light emitted from a source operating in the far infrared region of the spectrum could have significant energy (radiance), but an observer would hardly perceive it; its luminance would be almost zero.
- ***Brightness*** is a subjective descriptor of light perception that is practically impossible to measure. It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation.

Image Sensing and Acquisition

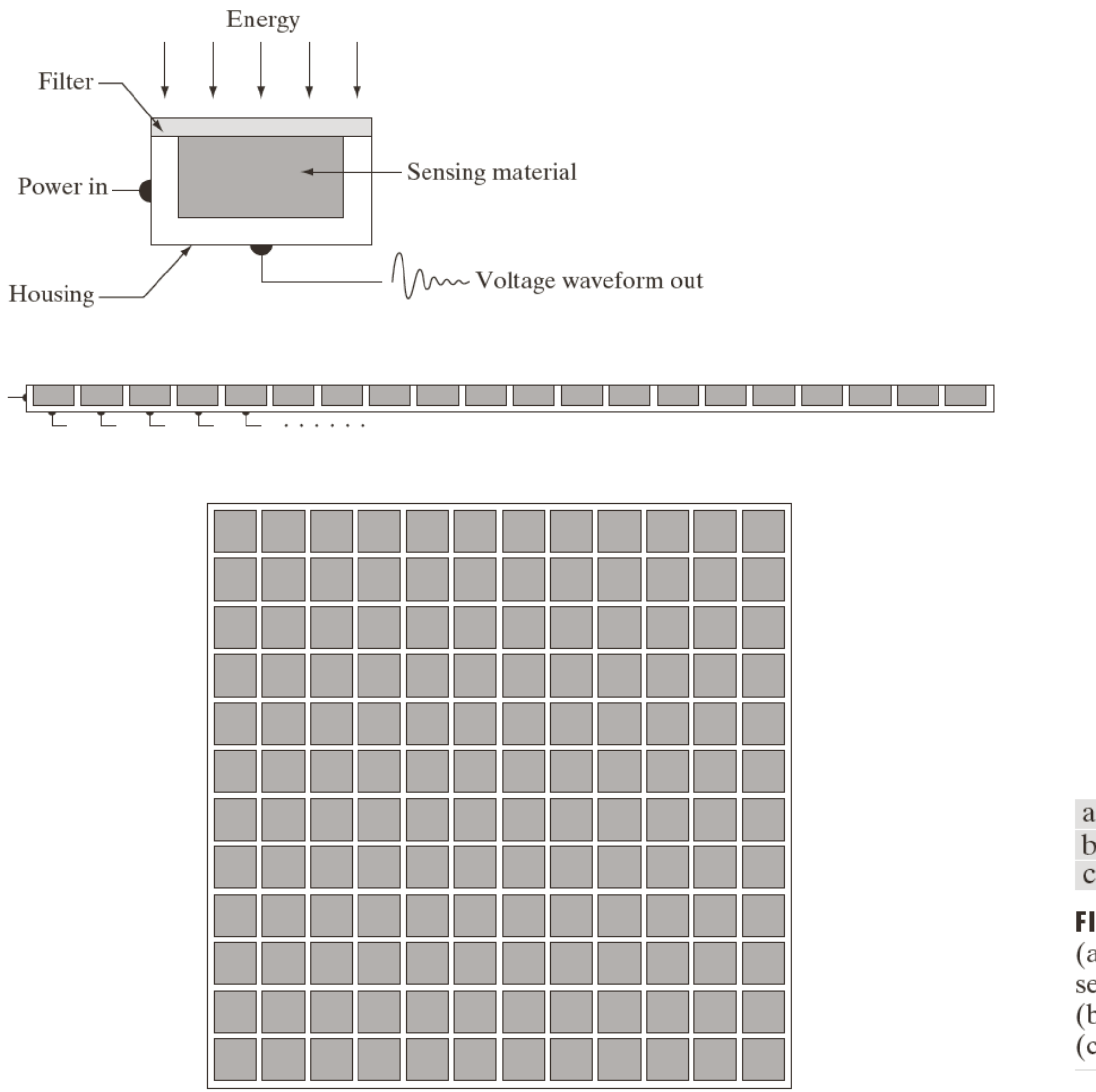


FIGURE 2.12
(a) Single imaging sensor.
(b) Line sensor.
(c) Array sensor.

Image Formation Model

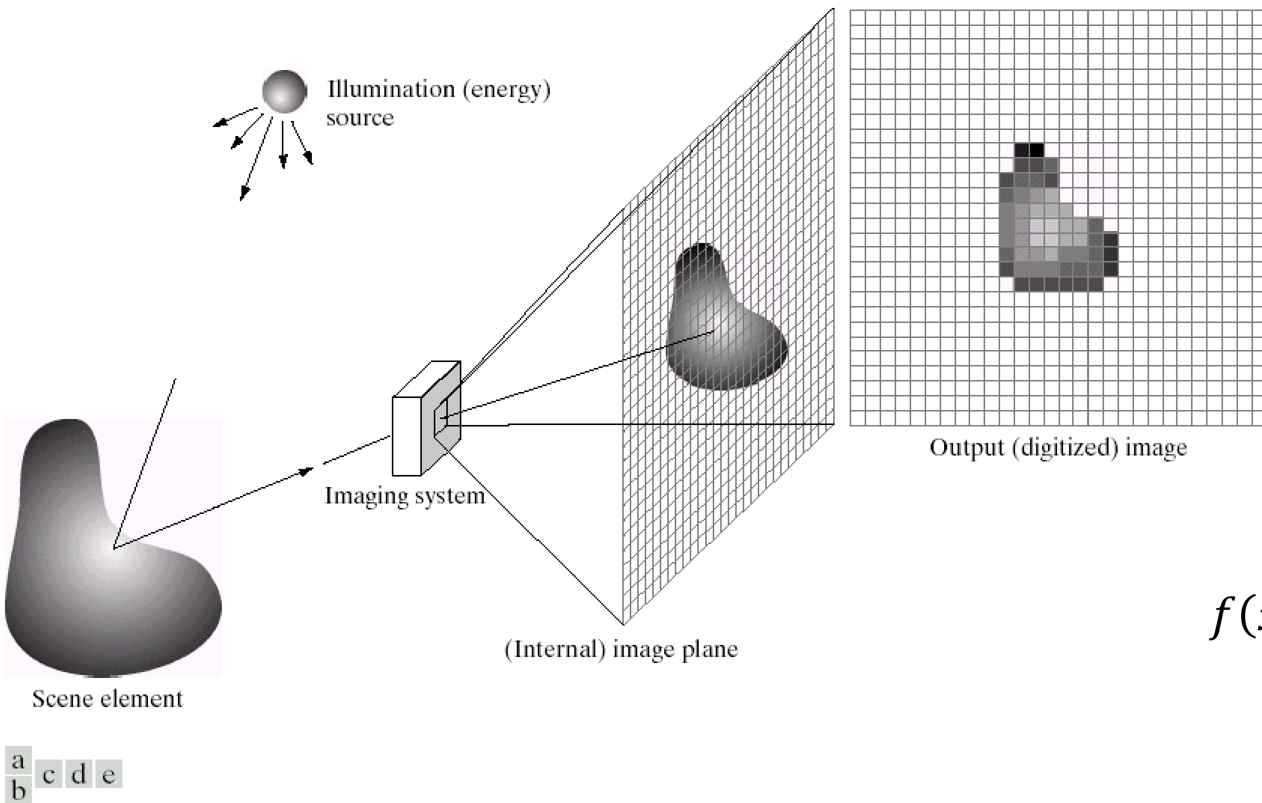


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

$$f(x, y) = i(x, y)r(x, y)$$
$$0 < i(x, y) < \infty$$
$$0 < r(x, y) < 1$$

Image Sampling and Quantization

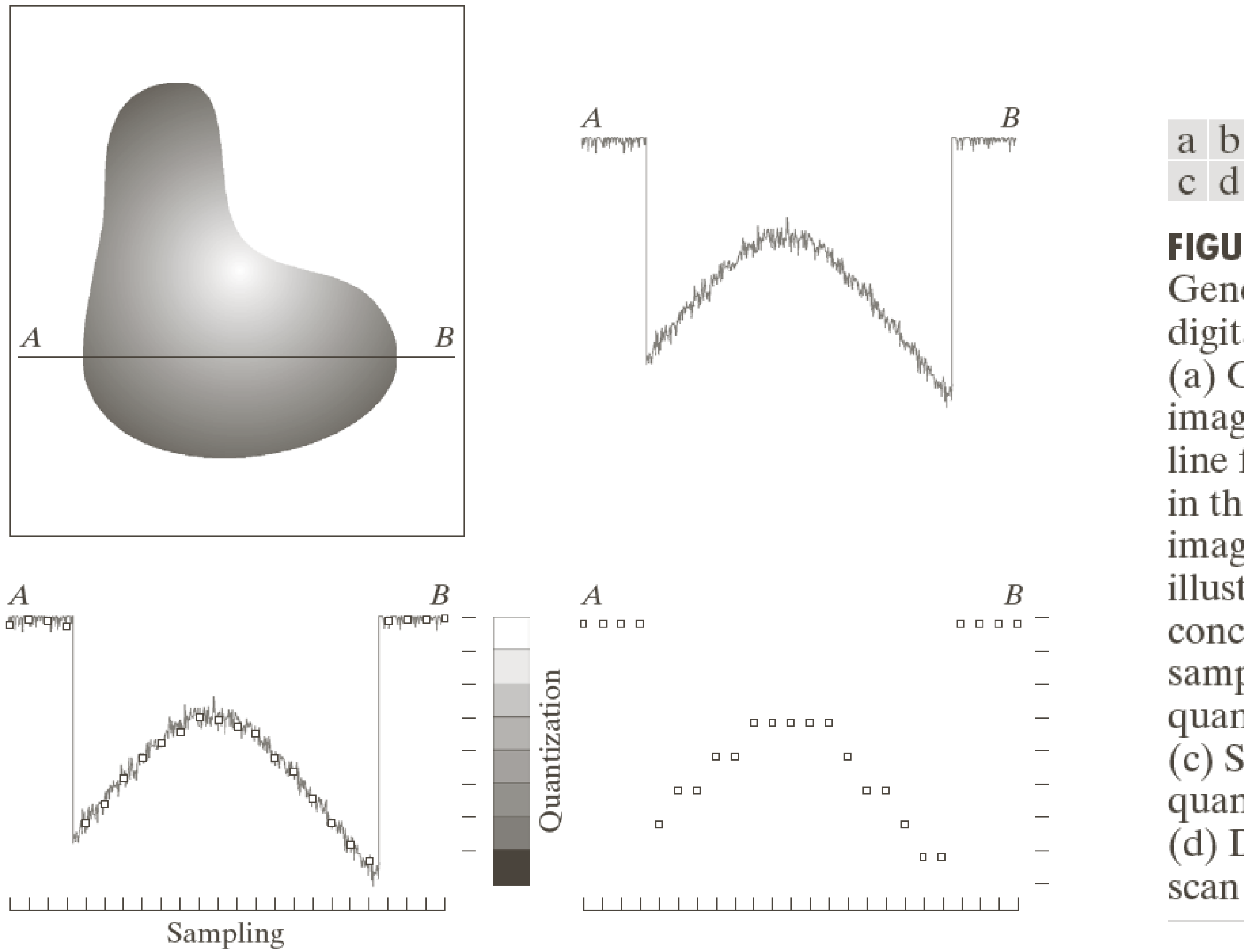
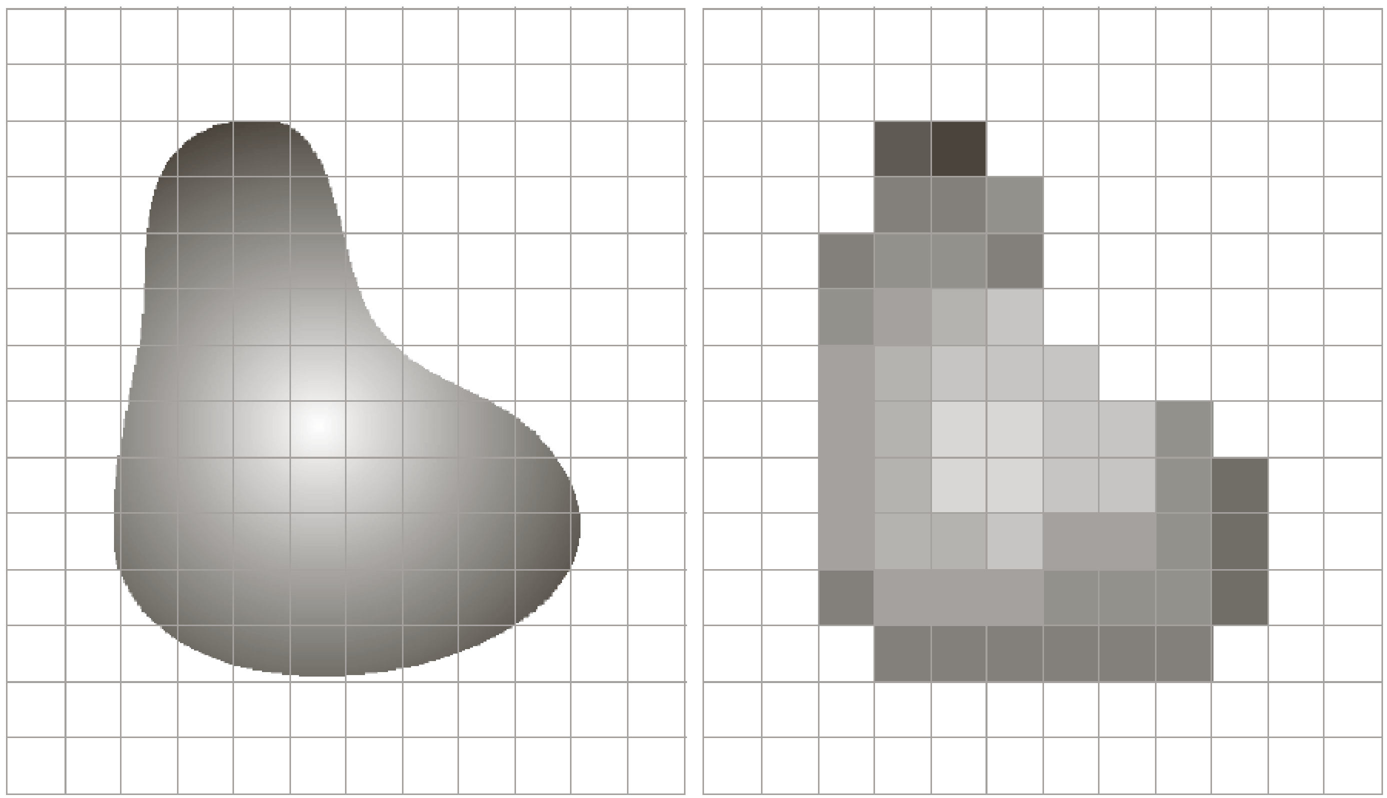


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

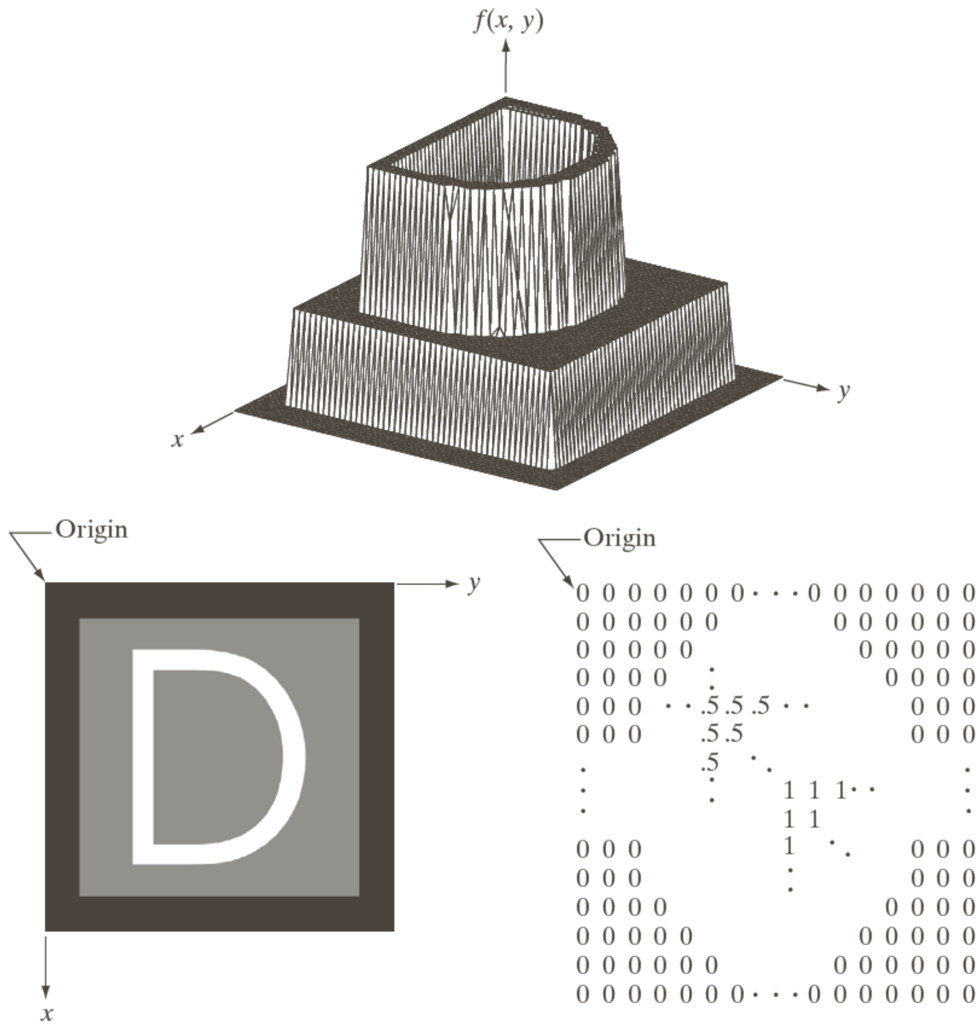


a b

FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Representing Digital Images

- Let $f(s,t)$ represent a continuous image function of two continuous variables, s and t. We convert this function into a digital image by sampling and quantization.
- Suppose that we sample the continuous image into a 2-D array, $f(x,y)$ containing M rows and N columns, where (x,y) are discrete coordinates.
- For notational clarity and convenience, we use integer values for these discrete coordinates: $x = 0, 1, 2, \ldots, M-1$ and $y = 0, 1, 2, \ldots, M-1$.



a
b c

FIGURE 2.18
(a) Image plotted as a surface.
(b) Image displayed as a visual intensity array.
(c) Image shown as a 2-D numerical array (0, .5, and 1 represent black, gray, and white, respectively).

Representations

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

Saturation and Noise

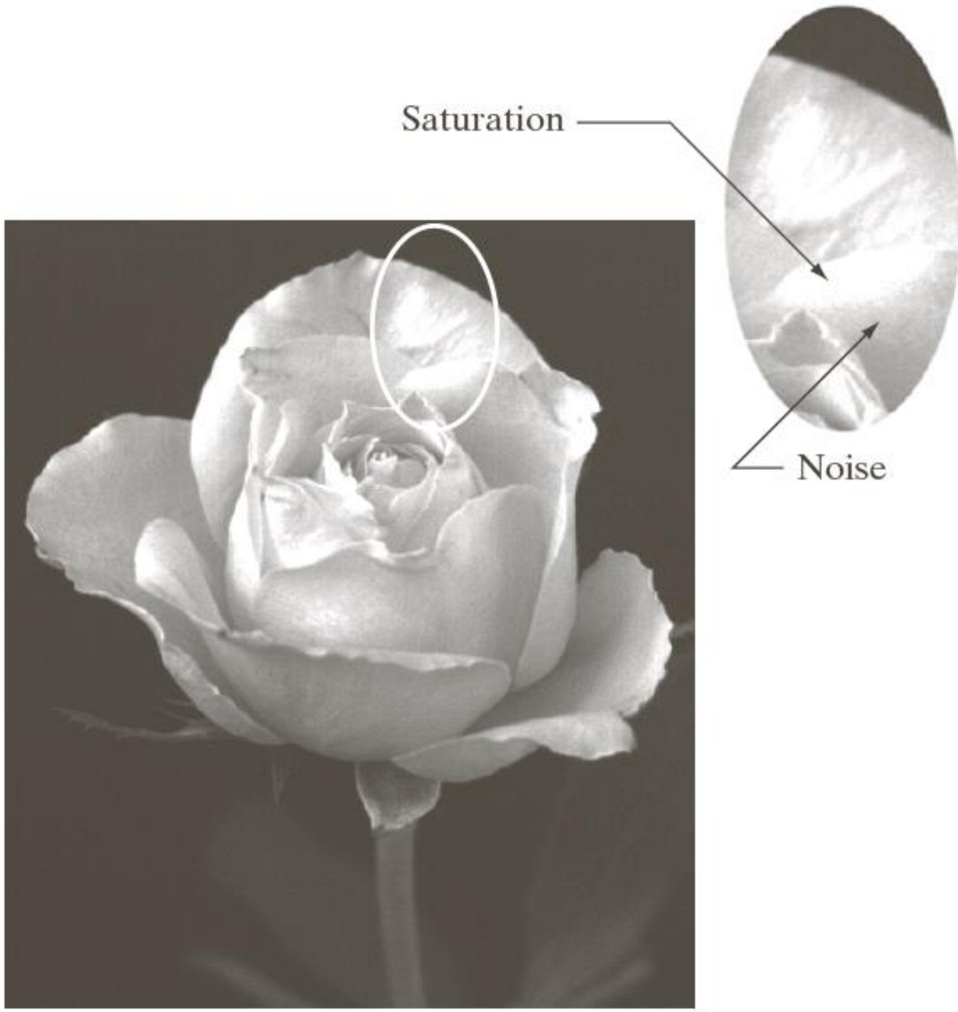


FIGURE 2.19 An image exhibiting saturation and noise. Saturation is the highest value beyond which all intensity levels are clipped (note how the entire saturated area has a high, *constant* intensity level). Noise in this case appears as a grainy texture pattern. Noise, especially in the darker regions of an image (e.g., the stem of the rose) masks the lowest detectable true intensity level.

$$L = 2^k$$

TABLE 2.1
Number of storage bits for various values of N and k.

N/k	1 (L = 2)	2 (L = 4)	3 (L = 8)	4 (L = 16)	5 (L = 32)	6 (L = 64)	7 (L = 128)	8 (L = 256)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

Spatial Resolution

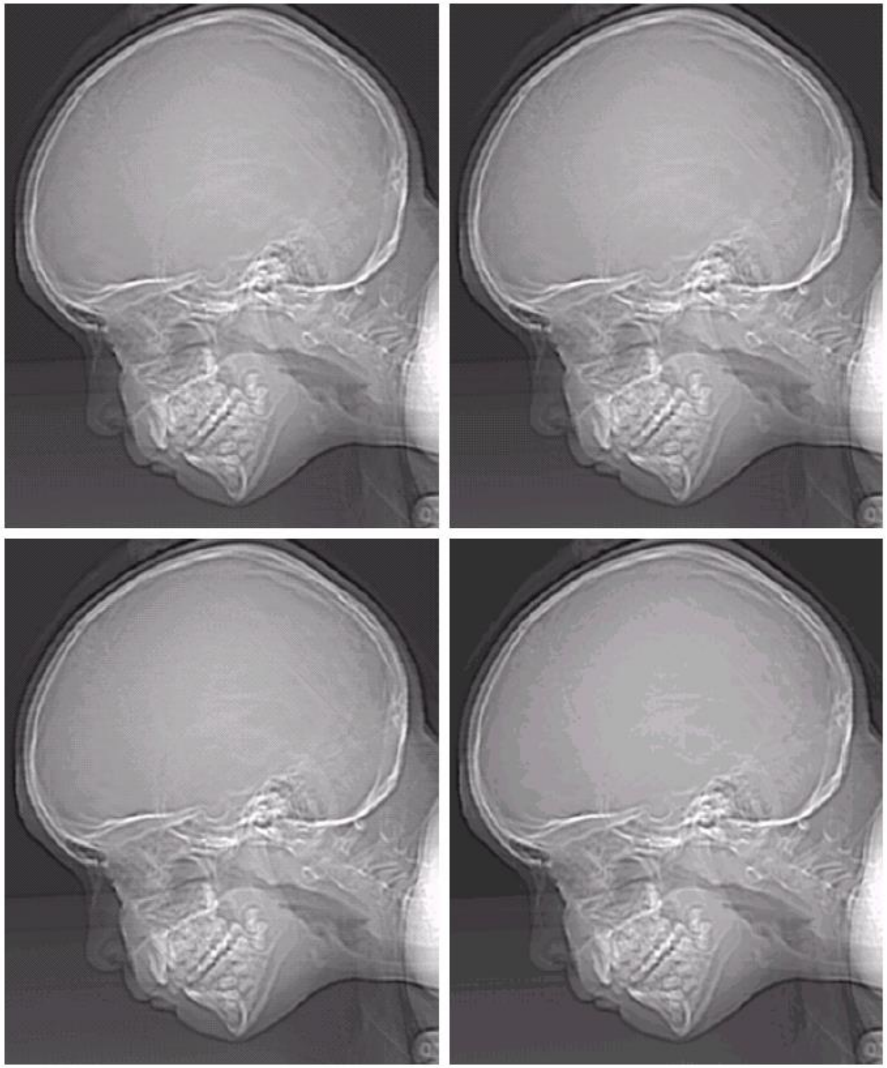
- Spatial Resolution is a measure of the smallest discernible detail in an image.
  - Quantitatively, *spatial resolution* can be stated in a number of ways, with *line pairs per unit distance,* and *dots (pixels) per unit distance* being among the most common measures.
- A widely used definition of image resolution is the largest number of *discernible* line pairs per unit distance (e.g., 100 line pairs per mm).
- Dots per unit distance is a measure of image resolution used commonly in the printing and publishing industry. In the U.S., this measure usually is expressed as *dots per inch* (dpi).
  - Newspapers are printed with a resolution of 75 dpi
  - Magazines at 133 dpi
  - Glossy brochures at 175 dpi
  - Text book page printed at 2400 dpi

Intensity Resolution

- Intensity resolution similarly refers to the smallest discernible change in intensity level.
- Based on hardware considerations, the number of intensity levels usually is an integer power of two.
- The most common number is 8 bits, with 16 bits being used in some applications in which enhancement of specific intensity ranges is necessary.
- Intensity quantization using 32 bits is rare.
- Sometimes one finds systems that can digitize the intensity levels of an image using 10 or 12 bits, but these are the exception, rather than the rule.

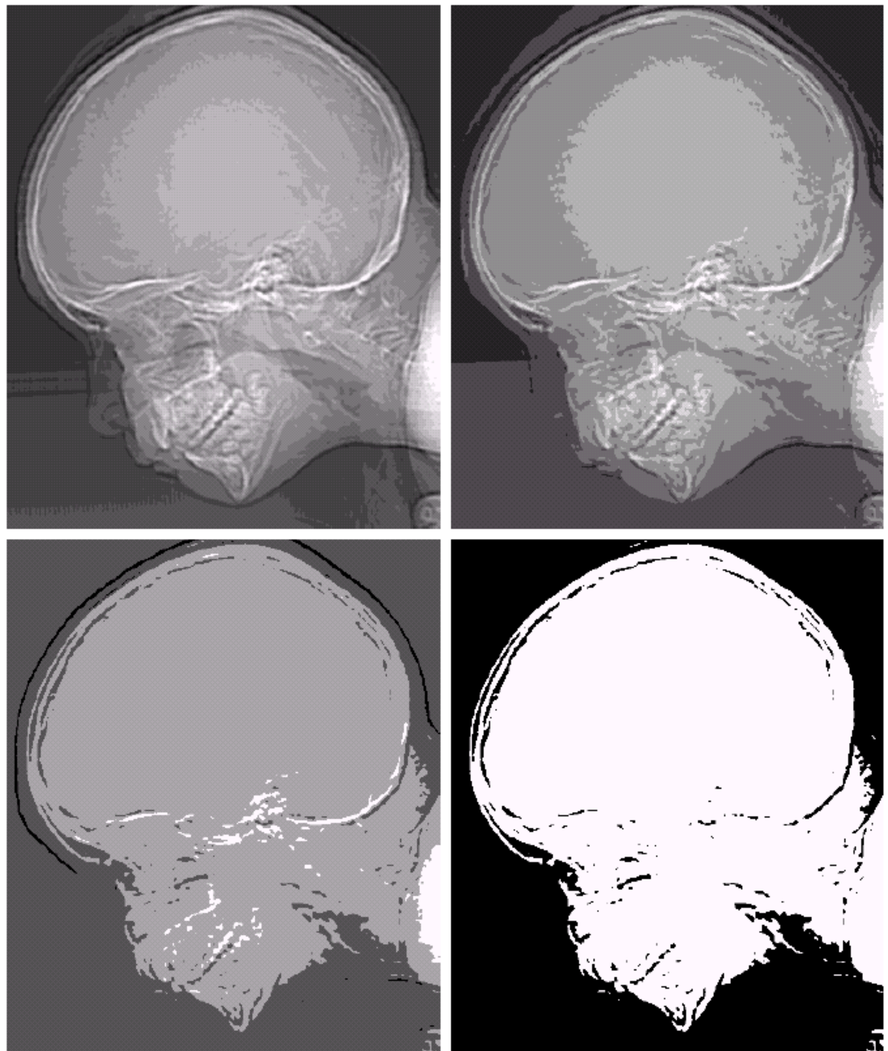
a b
c d

FIGURE 2.20 Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.



a b
c d

FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.



e	f
g	h

FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

Image Subjective Quality

a b c

FIGURE 2.22 (a) Image with a low level of detail. (b) Image with a medium level of detail. (c) Image with a relatively large amount of detail. (Image (b) courtesy of the Massachusetts Institute of Technology.)

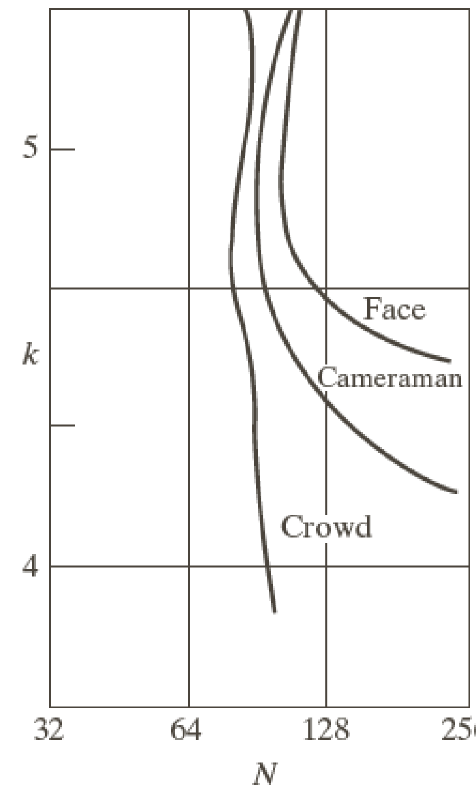


FIGURE 2.23 Typical isopreference curves for the three types of images in Fig. 2.22.

Image Interpolation

- Interpolation is a basic tool used extensively in tasks such as zooming, shrinking, rotating, and geometric corrections.
- We given an introduction to interpolation and apply it to image resizing (shrinking and zooming), which are basically image resampling methods.
- *Interpolation* is the process of using known data to estimate values at unknown locations.
  - Example: An image of size 500 X 500 pixels has to be enlarged 1.5 times to 750 X 750 pixels

Methods

- Nearest neighbor interpolation
  - Assigns to each new location the intensity of its nearest neighbor in the original image
  - It has the tendency to produce undesirable artifacts, such as severe distortion of straight edges.
- *Bilinear interpolation*
  - Use the four nearest neighbors to estimate the intensity at a given location: $v(x,y) = ax + by + cxy + d$
- *Bicubic interpolation*
  - Involves the sixteen nearest neighbors of a point.

$$v(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j$$

a	b	c
d	e	f

FIGURE 2.24 (a) Image reduced to 72 dpi and zoomed back to its original size (3692 × 2812 pixels) using nearest neighbor interpolation. This figure is the same as Fig. 2.20(d). (b) Image shrunk and zoomed using bilinear interpolation. (c) Same as (b) but using bicubic interpolation. (d)–(f) Same sequence, but shrinking down to 150 dpi instead of 72 dpi [Fig. 2.24(d) is the same as Fig. 2.20(c)]. Compare Figs. 2.24(e) and (f), especially the latter, with the original image in Fig. 2.20(a).

Basic Relations Between Pixels

- Neighbors of a pixel
  - Four horizontal and vertical neighbors -- $N_4(p)$

    $(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$
  - Four diagonal neighbors -- $N_D(p)$

    $(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$
  - 8-neighbors of p -- $N_8(p)$
    - Combination of $N_4(p)$ and $N_D(p)$

$N_4(\mathbf{p})$

	(x-1,y)	
(x,y-1)	(x,y)	(x,y+1)
	(x+1,y)	

$N_D(\mathbf{p})$

(x-1,y-1)		(x-1,y+1)
	(x,y)	
(x-1,y+1)		(x+1,y+1)

$N_8(\mathbf{p})$

(x-1,y-1)	(x-1,y)	(x-1,y+1)
(x,y-1)	(x,y)	(x,y+1)
(x-1,y+1)	(x+1,y)	(x+1,y+1)

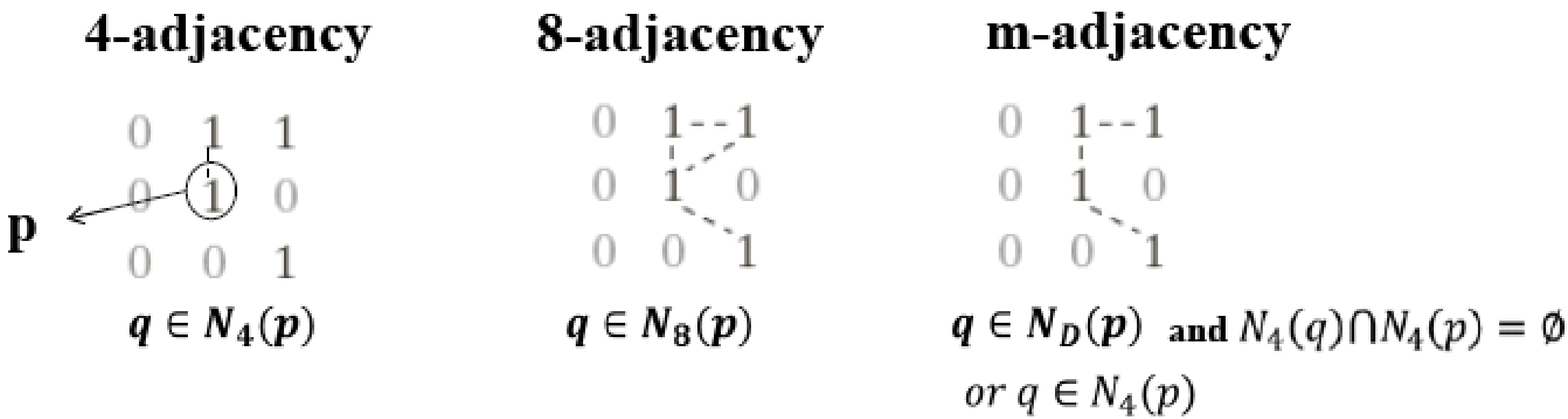
Adjacency

- Let V be the set of intensity values used to define adjacency.
  - In a binary image, $V = \{1\}$ if we are referring to adjacency of pixels with value 1.
  - In a gray-scale image, set V typically contains more elements. For example, in the adjacency of pixels with a range of possible intensity values 0 to 255, set V could be any subset of these 256 values.

Three Types of Adjacency

- *4-adjacency*
  - Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$.
- *8-adjacency*
  - Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(p)$.
- *m-adjacency* (mixed adjacency).
  - Two pixels p and q with values from V *are* m-adjacent if
    - q is in $N_4(p)$*, or*
    - q is in $N_D(p)$ *and* the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

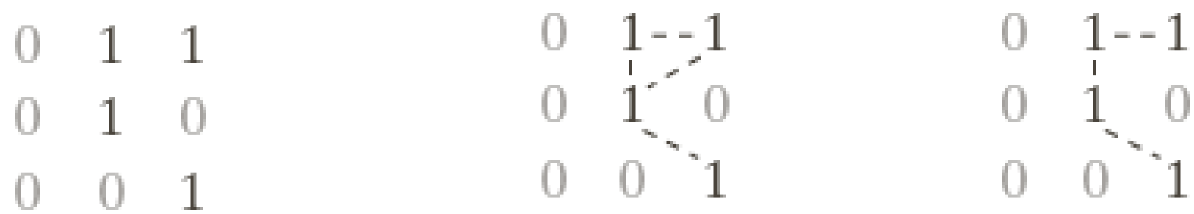
Remove the Ambiguity of 8-Adjacency



- m-adjacency has eliminated the multiple path connection that has been generated by the 8-adjacency.

Digital Path

- A digital path (or curve) from pixel p with coordinate (x, y) to pixel q with coordinate (s, t) is a sequence of distinct pixels with coordinates $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ where $(x_0, y_0) = (x, y)$ and $(x_n, y_n) = (s, t)$ and pixels (x_i, y_i) and (x_{i-1}, y_{i-1}) are adjacent for $1 \leq i \leq n$.
- n is the length of the path.
- If $(x_0, y_0) = (x_n, y_n)$, the path is closed.
- We can specify 4-, 8- or m-paths depending on the type of adjacency specified.



a b c

In figure (b) the paths between the top right and bottom right pixels are 8-paths. And the path between the same 2 pixels in figure (c) is m-path.

Connectivity

- Let S represent a subset of pixels in an image. Two pixels p and q are said to be connected in S if there exists a path between them consisting entirely of pixels in S.
- For any pixel p in S, the set of pixels that are connected to it in S is called a ***connected component*** of S.
- If it only has one connected component, then set S is called a *connected set*.

Region and Boundary

- **Region**
  - Let R be a subset of pixels in an image, we call R a region of the image if R is a connected set.
- **Boundary**
  - The *boundary* (also called *border* or *contour*) of a region *R* is the set of pixels in the region that have one or more neighbors that are not in R.

Region and Boundary

- If R happens to be an entire image, then its boundary is defined as the set of pixels in the first and last rows and columns in the image.
- This extra definition is required because an image has no neighbors beyond its borders.
- Normally, when we refer to a region, we are referring to subset of an image, and any pixels in the boundary of the region that happen to coincide with the border of the image are included implicitly as part of the region boundary.

Distance Measures

- For pixels p, q, and z, with coordinates $(x, y), (s, r)$, and (v, w), respectively, D is a *distance function* or *metric* if
  - $D(p,q) \geq 0$ ($D(p,q) = 0$ iff $p = q$),
  - $D(p,q) = D(q,p)$, and
  - $D(p,z) = D(p,q) + D(q,z)$.
- Euclidean distance:

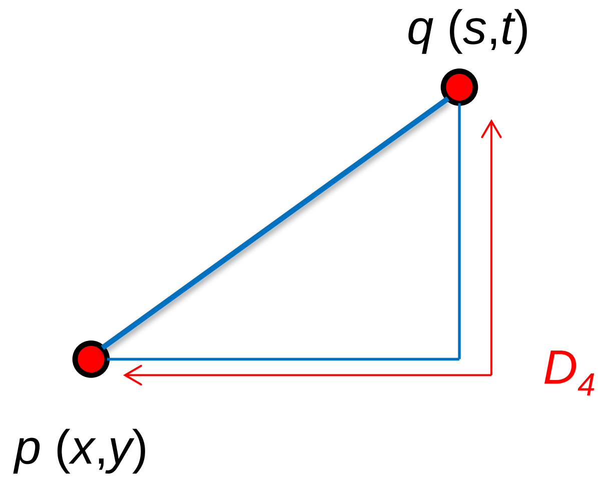
$$D_e(p,q) = [(x-s)^2 + (y-t)^2]^{\frac{1}{2}}$$

D_4 Distance

- The D_4 *distance* (also called *city-block distance*) between *p* and *q* is defined as:

$$D_4(p, q) = |x - s| + |y - t|$$

- Pixels having a D_4 distance from (x, y), less than or equal to some value r form a diamond centered at (x, y).
- The pixels with $D_4 = 1$ are the 4-neighbors of (x, y).



		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

D_8 Distance

- D_8 distance (called the chessboard distance) between p and q is defined as
$$D_8(p,q) = \max(|x-s|, |y-t|)$$
- The pixels with D_8 distance from (x, y) less than or equal to some value r form a square centered at (x, y).
- For example, $r = 2$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

- The pixels with $D_8 = 1$ are the 8-neighbors of (x, y).

D_m Distance

- $\boldsymbol{D_m}$ **distance** is defined as the shortest m-path between the points.
- In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.

Example

- Consider the following arrangement of pixels and assume that p, p_2, and p_4 have value 1 and that p_1 and p_3 can have can have a value of 0 or 1

  Suppose that we consider

  the adjacency of pixels

  values 1 (i.e. $V = \{1\}$)

	p_3	p_4
p_1	p_2	
p		

Case 1

- Now, to compute the D_m between points p and p_4

  Here we have 4 cases:

  Case1: If $p_1 = 0$ and $p_3 = 0$

  The length of the shortest m-path

  (the D_m *distance) is 2 (*p, p_2, p_4*)*

	0	1
0	1	
1		

	p_3	p_4
p_1	p_2	
p		

Case 2

- **Case2:** If p_1 =1 and p_3 = 0

  now, p_2 *and* p *will no longer be adjacent (see m-adjacency definition)*

  then, the length of the shortest

  path will be 3 (p, p_1, p_2, p_4)

	0	1
1	1	
1		

	p_3	p_4
p_1	p_2	
p		

Cases 3 & 4

	p_3	p_4
p_1	p_2	
p		

- **Case3:** If $p_1 = 0$ and $p_3 = 1$

  The same applies here, and the shortest –m-path will be 3 (p, p_2, p_3, p_4)

	1	1
0	1	
1		

- **Case4:** If $p_1 = 1$ and $p_3 = 1$

  The length of the shortest m-path will be 4 (p, p_1 , p_2, p_3, p_4)

	1	1
1	1	
1		

Mathematical Tools

- Array vs. Matrix Operations
  - Array operations involving one or more images are carried out on a pixel-by-pixel basis

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

  - There are many situations where operations between images are carried using matrix operations.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Linear Operations

$$H\left[f(x, y)\right] = g(x, y)$$

$$H\left[a_i f_i(x, y) + a_j f_j(x, y)\right] = a_i H\left[f_i(x, y)\right] + a_j H\left[f_j(x, y)\right]$$

$$= a_i g_i(x, y) + a_j g_j(x, y)$$

$$\sum\left[a_i f_i(x, y) + a_j f_j(x, y)\right] = \sum a_i f_i(x, y) + \sum a_j f_j(x, y)$$

$$= a_i \sum f_i(x, y) + a_j \sum f_j(x, y)$$

$$= a_i g_i(x, y) + a_j g_j(x, y)$$

Non-Linear Operations

$$f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \quad \text{and} \quad f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$$

$$\max\left\{(1)\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1)\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = \max\left\{\begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix}\right\}$$

$$= -2$$

$$(1)\max\left\{\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}\right\} + (-1)\max\left\{\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = 3 + (-1)7$$

$$= -4$$

Arithmetic Operations

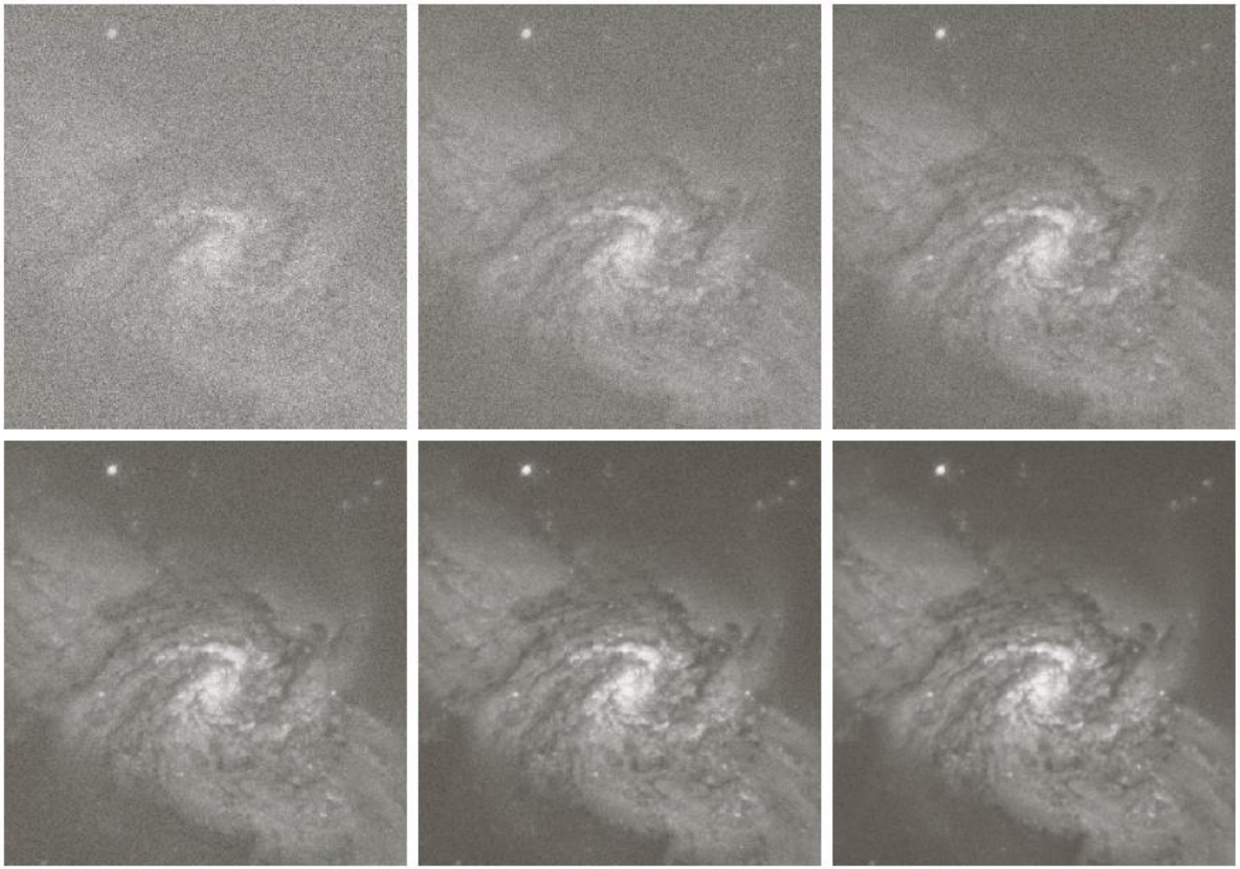
$$s(x, y) = f(x, y) + g(x, y)$$

$$d(x, y) = f(x, y) - g(x, y)$$

$$p(x, y) = f(x, y) \times g(x, y)$$

$$v(x, y) = f(x, y) \div g(x, y)$$

De-Noising



a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

$$g(x, y) = f(x, y) + \eta(x, y)$$

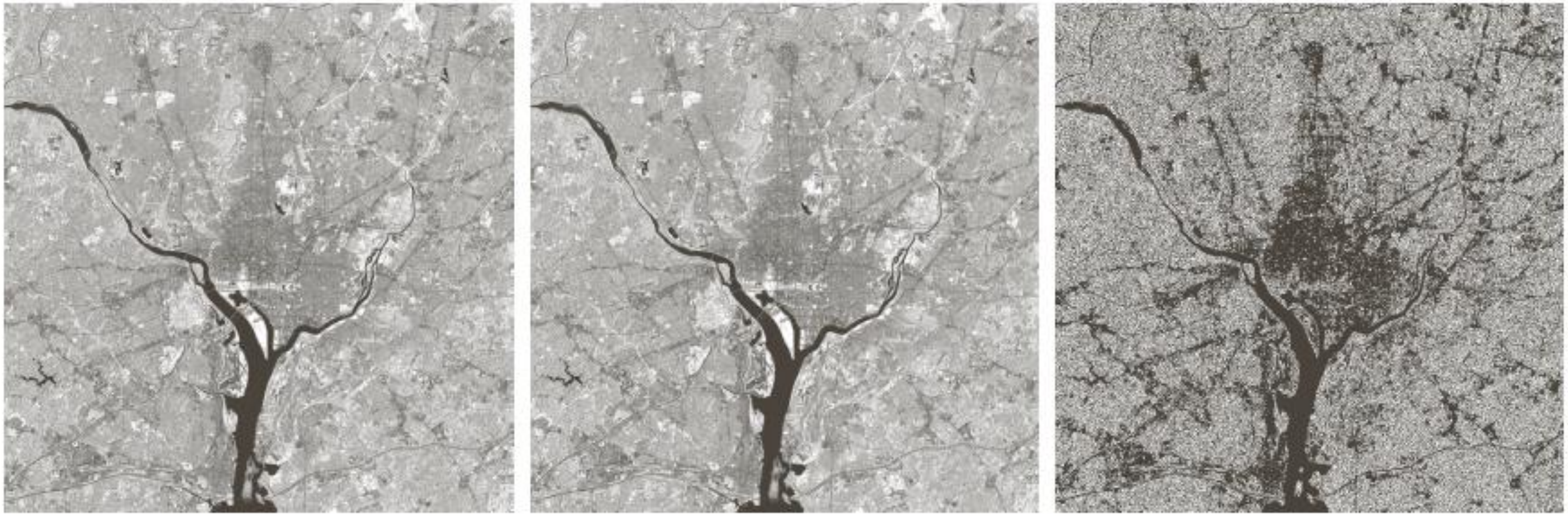
$$\bar{g}(x, y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x, y)$$

$$E\{\bar{g}(x, y)\} = f(x, y)$$

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$$

Assumption: Noise is uncorrelated to image and has zero mean.

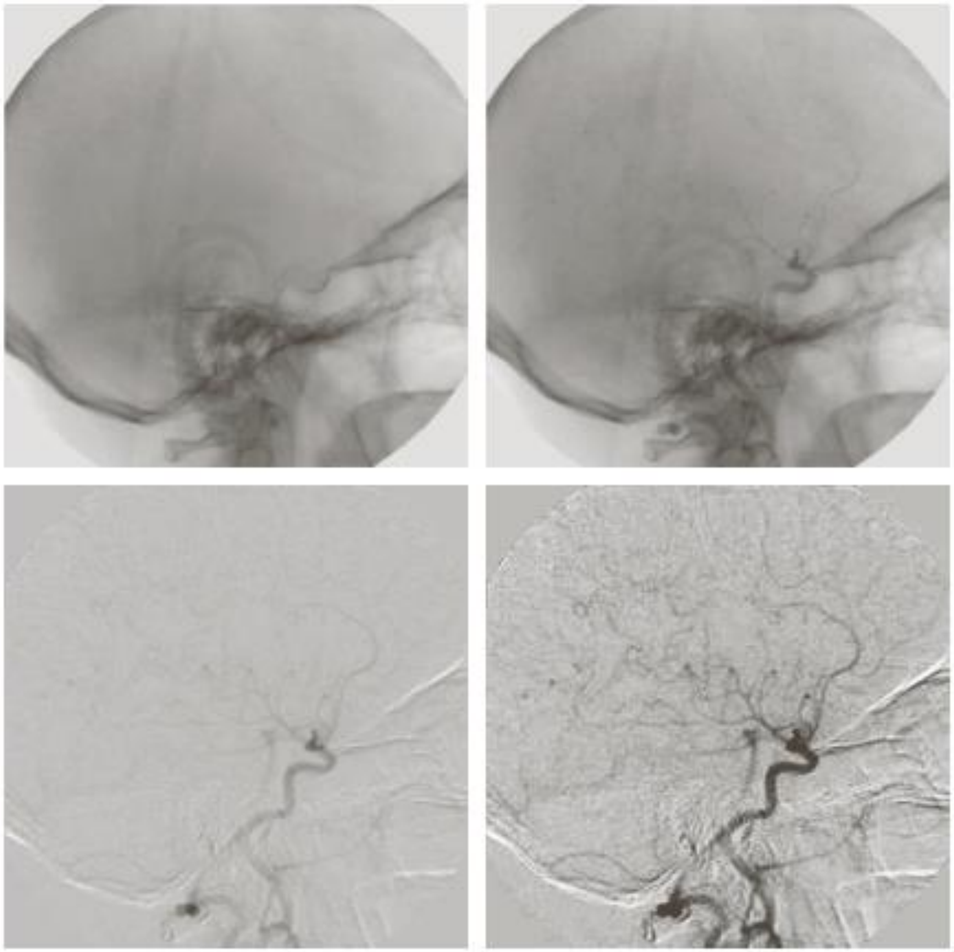
Image Subtraction – Enhance Difference



a b c

FIGURE 2.27 (a) Infrared image of the Washington, D.C. area. (b) Image obtained by setting to zero the least significant bit of every pixel in (a). (c) Difference of the two images, scaled to the range [0, 255] for clarity.

Mask Mode Radiography



a	b
c	d

FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

Image Division

$$g(x, y) = f(x, y)h(x, y)$$

a b c

FIGURE 2.29 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

ROI Masking

a b c

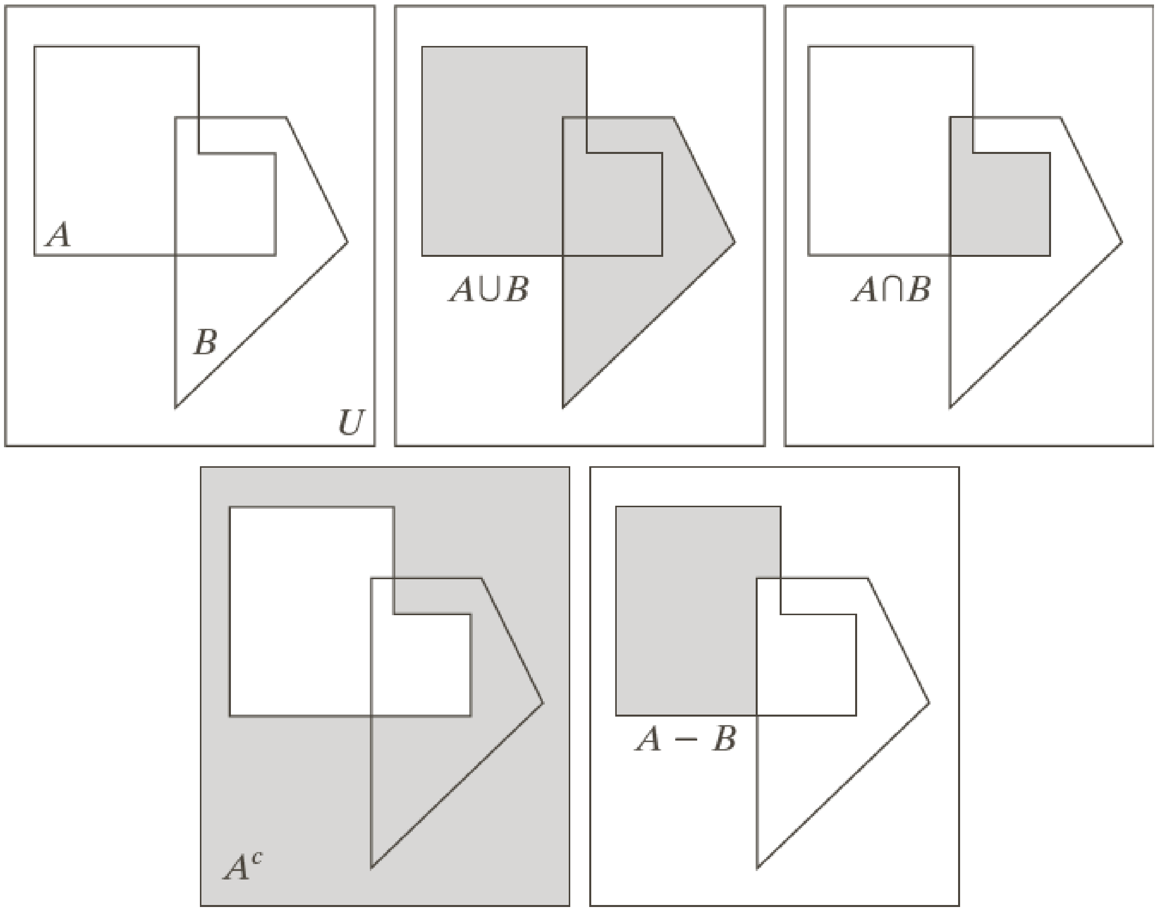
FIGURE 2.30 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

Notes on Arithmetic Operations

- The images used in averaging and subtraction must be registered.
- Output images should be normalized to the range of $[0, 255]$.

$$f_m = f - \min(f)$$
$$f_s = K[f_m / \max(f_m)]$$

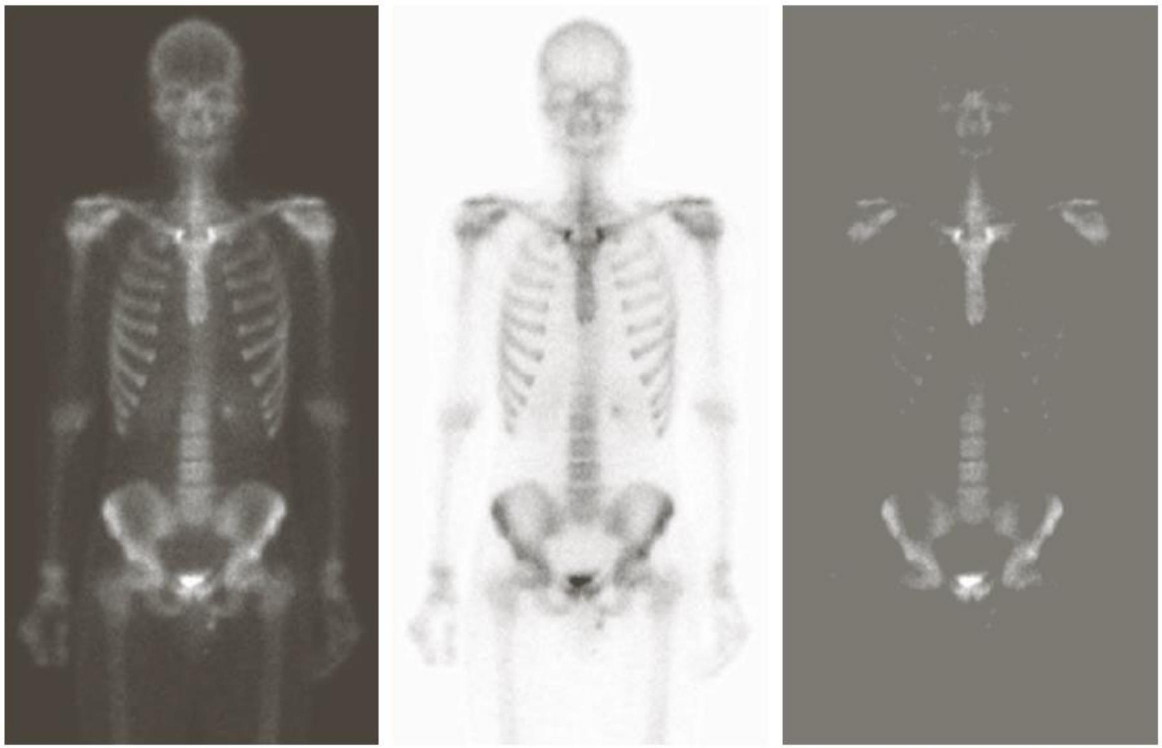
Set Operations



a b c
d e

FIGURE 2.31
(a) Two sets of coordinates, A and B, in 2-D space. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B. In (b)–(e) the shaded areas represent the member of the set operation indicated.

Union of Gray-scale Sets



a b c

FIGURE 2.32 Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

$A^c = \{(x, y, K - z) \mid (x, y, z) \in A\}$ **Complement – negative image**

$A \cup B = \{(x, y, \max(z_a, z_b)) \mid (x, y, z_a) \in A, (x, y, z_b) \in B\}$

Logical Operations for Binary Images

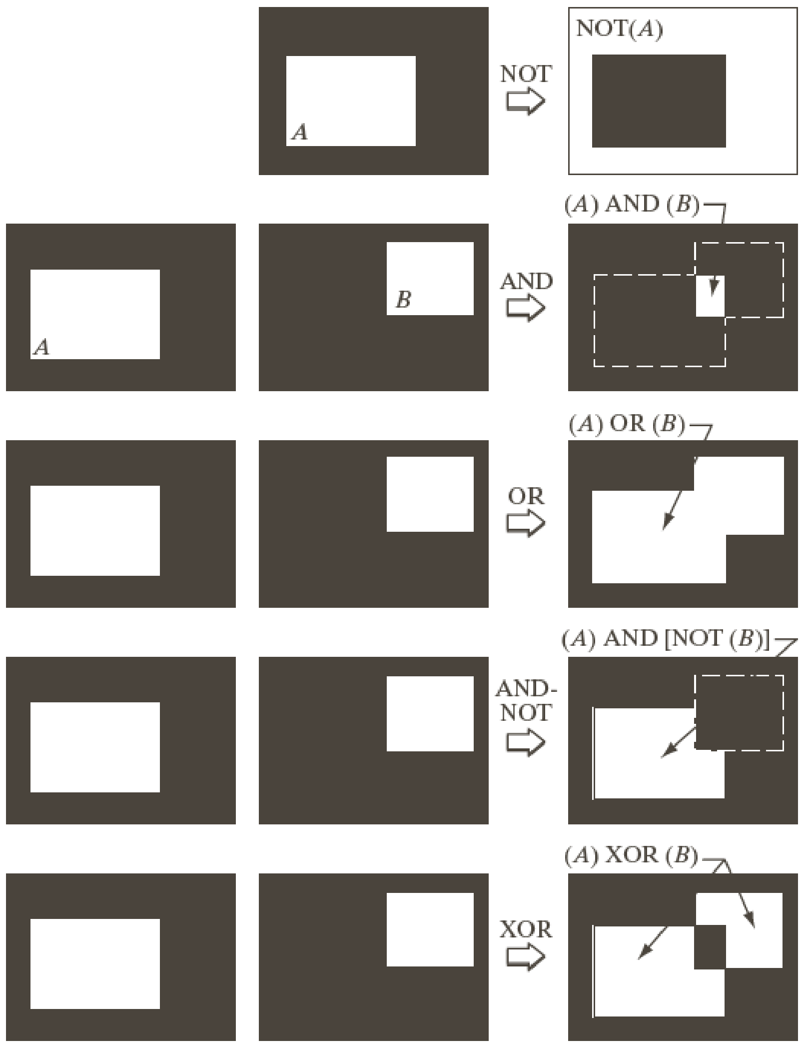


FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Spatial Operations

- Spatial operations are performed directly on the pixels of a given image. We classify spatial operations into three broad categories:
  - Single-pixel operations
    - $s = T(z)$, where z is the intensity of a pixel in the original image and s is the (mapped) intensity of the corresponding pixel in the processed image.
  - Neighborhood operations
    - Generate a corresponding pixel at the same coordinates in an output (processed) image, such that the value of that pixel is determined by a specified operation involving the pixels in a neighborhood of the input image
  - Geometric spatial transformations
    - Analogous to "printing" an image on a sheet of rubber and then stretching the sheet according to a predefined set of rules.

Single-Pixel Operation

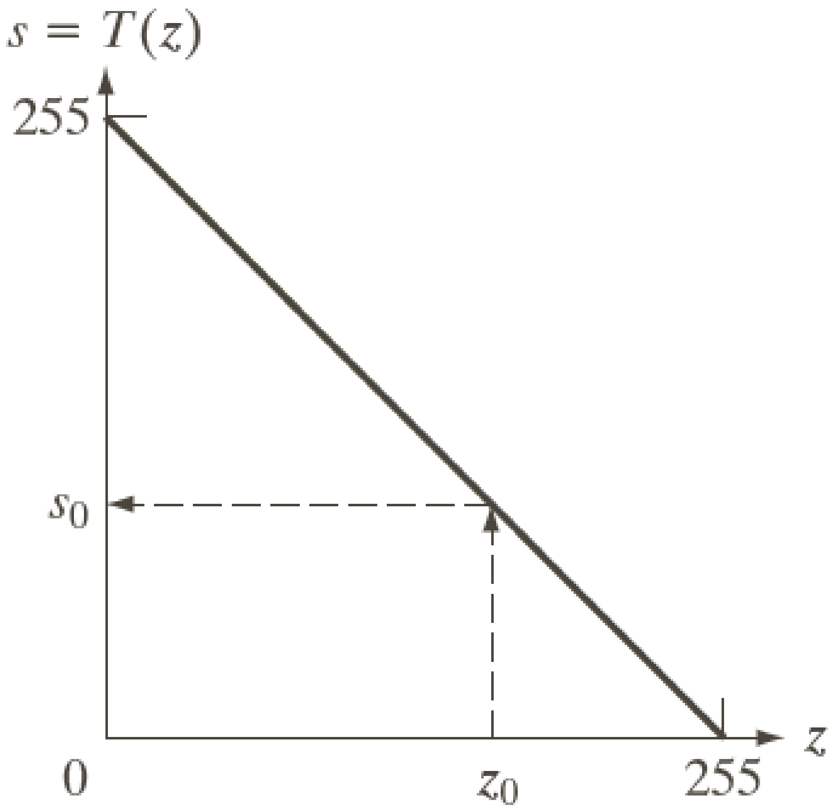
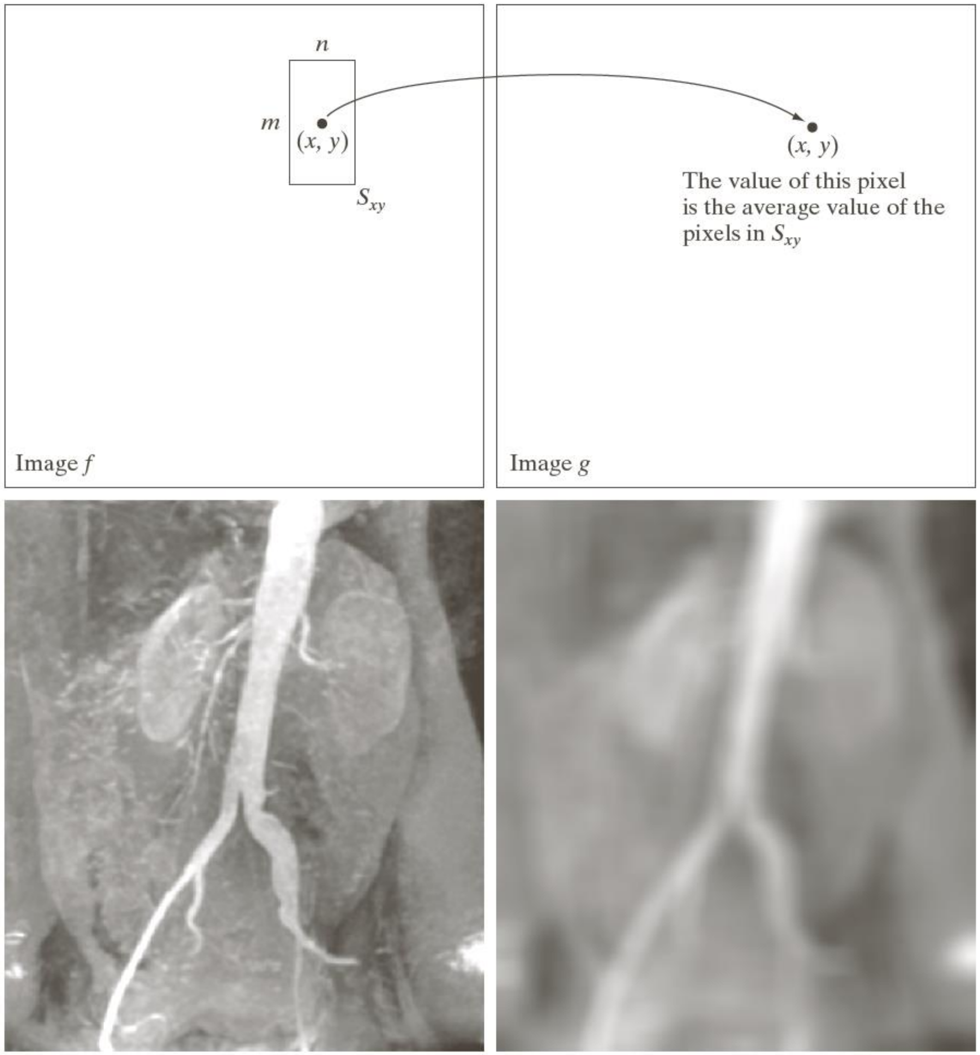


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.

$$g(x, y) = \frac{1}{mn} \Sigma_{(r,c) \in S_{xy}} f(r, c)$$



a b
c d

FIGURE 2.35 Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram discussed in Section 1.3.2. (d) The result of using Eq. (2.6-21) with $m = n = 41$. The images are of size 790×686 pixels.

Geometric Spatial Transformation

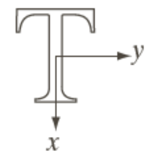
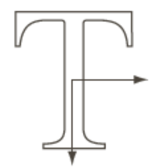
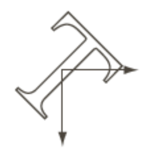
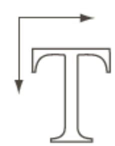
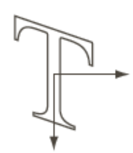
- A geometric transformation consists of two basic operations:
  - A spatial transformation of coordinates $(x, y) = T\{(v, w)\}$ where (v, w) are pixel coordinates in the original image, and (x, y) are the corresponding pixel coordinates in the transformed image.
    - Examples

      Shrink the original image to half its size

      $(x, y) = T\{(v, w)\} = (v/2, w/2)$
  - Intensity interpolation that assigns intensity values to the spatially transformed pixels.

$$[x \;\; y \;\; 1] = [v \;\; w \;\; 1]\,\mathbf{T} = [v \;\; w \;\; 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

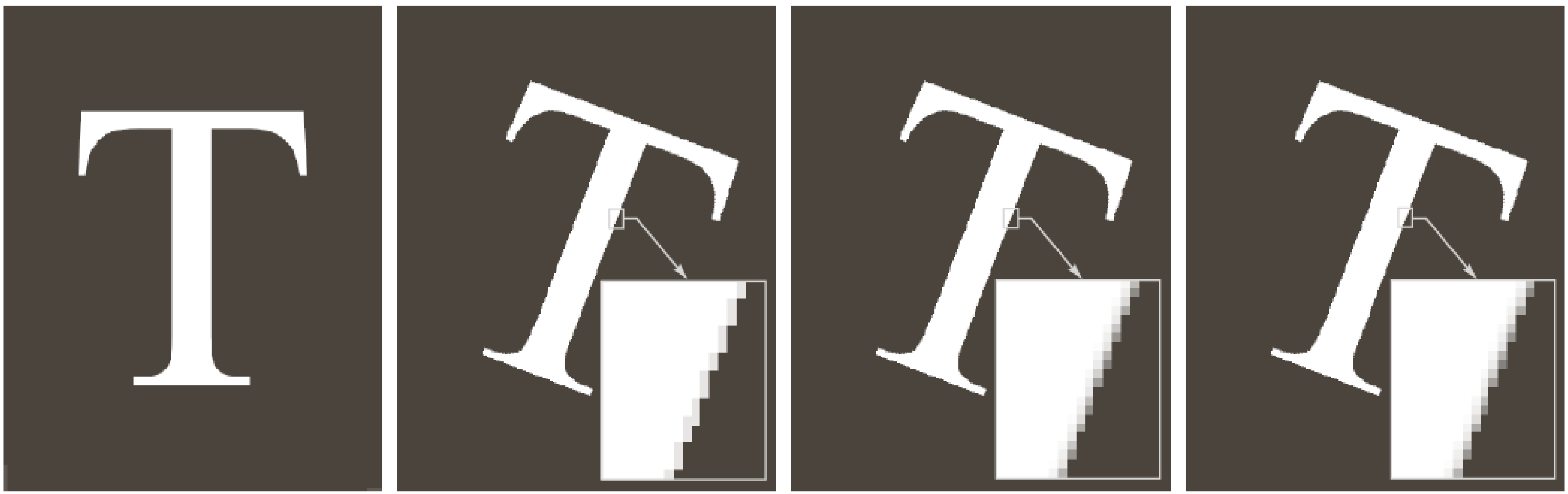
TABLE 2.2
Affine transformations based on Eq. (2.6.–23).

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\sin\theta + w\cos\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Forward vs. Inverse Mapping

- Forward mapping $(x, y) = T\{(v, w)\}$
  - Scanning the pixels of the input image and. At each location, (v, w), computing the spatial location, (x, y), of the corresponding pixel in the output image.
  - A problem with the forward mapping approach is that two or more pixels in the input image can be transformed to the same location in the output image, raising the question of how to combine multiple output values into a single output pixel.
  - In addition, it is possible that some output locations may not be assigned a pixel at all.
- Inverse mapping
  - Scanning the output pixel locations and, at each location, (x, y), computes the corresponding location in the input image using $(v, w) = T^{-1}(x, y)$.
  - It then interpolates (using one of the techniques discussed previously among the nearest input pixels to determine the intensity of the output pixel value.
  - Inverse mappings are more efficient to implement than forward mappings and are used in numerous commercial implementations of spatial transformations

Rotation and Intensity Interpolation



a b c d

FIGURE 2.36 (a) A 300 dpi image of the letter T. (b) Image rotated 21° clockwise using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated 21° using bilinear interpolation. (d) Image rotated 21° using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.

Image Registration

- Image registration is an important application of digital image processing used to align two or more images of the same scene.
- In the preceding discussion, the form of the transformation function required to achieve a desired geometric transformation was known.
- In image registration, we have available the input and output images, but the transformation that the output image from the input is unknown.
- The problem then is to estimate the transformation function and then use it to register the two images.
- To clarify terminology, the **input image** is the image we wish to transform, and what we call the **reference image** is the image, against which we want to register the input.

Tie Points

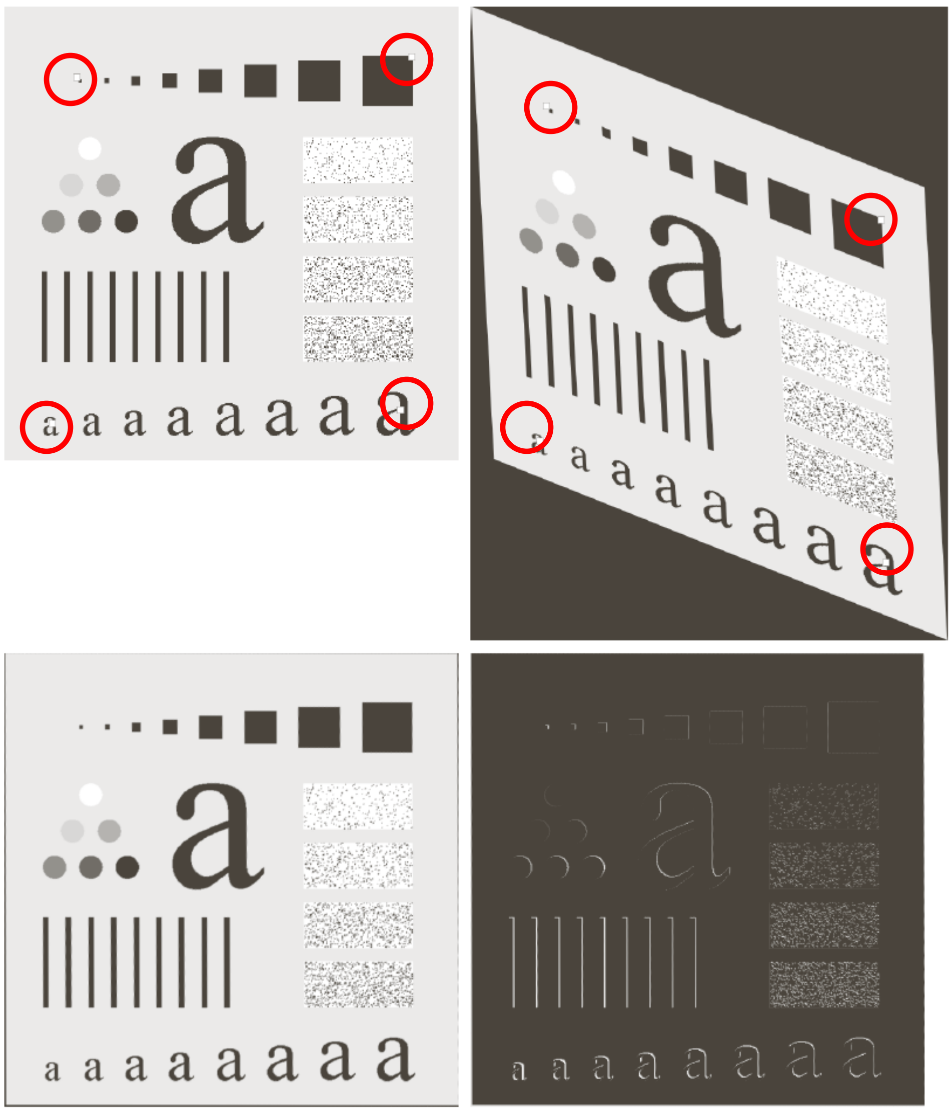
- One of the principal approaches for solving the problem is to use
  tie points (also called **control points**), which are corresponding points whose locations are known precisely in the input and reference images.
- There are numerous ways to select tie points, ranging from interactively selecting them to applying algorithms that attempt to detect these points automatically.
- In some applications, imaging systems have physical artifacts (such as small metallic objects) embedded in the imaging sensors. These produce a set of known points (called **Réseau Marks**) directly on all images captured by the system, which can be used as guides for establishing tie points.

Transformation Function Estimation

- The problem of estimating the transformation function is one of modeling.
- For example, suppose that we have a set of four tie points each in an input and a reference image. A simple model based on a bilinear approximation is given by

$$x = c_1 v + c_2 w + c_3 vw + c_4$$
$$y = c_5 v + c_6 w + c_7 vw + c_8$$

  where, during the estimation phase, (v, w) and (x, y) are the coordinates of tie points in the input and reference images, respectively
- If we have four pairs of corresponding tie points in both images, we can write eight equations and use them to solve for the eight unknown coefficients.



a	b
c	d

FIGURE 2.37 Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the borders). (d) Difference between (a) and (c), showing more registration errors.

Vector and Matrix Operations

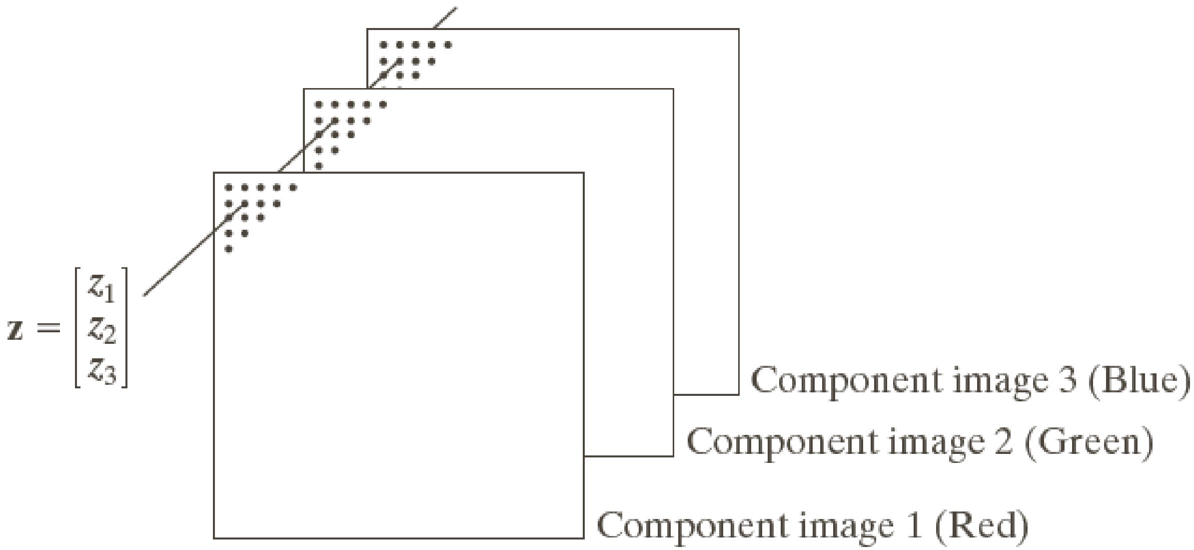


FIGURE 2.38 Formation of a vector from corresponding pixel values in three RGB component images.

Euclidean Distance (also called vector norm $||z - a||$):

$$D(\mathbf{z}, \mathbf{a}) = \left[(\mathbf{z} - \mathbf{a})^T(\mathbf{z} - \mathbf{a})\right]^{\frac{1}{2}}$$

$$= \left[(z_1 - a_1)^2 + (z_2 - a_2)^2 + \cdots + (z_n - a_n)^2\right]^{\frac{1}{2}}$$

Pixel and Image Vectors

- Linear transformation $\mathbf{w} = \mathbf{A}(\mathbf{z} - \mathbf{a})$
- we can express an image of size $M \times N$ as a vector of dimension $MN \times 1$ by letting the first row of the image be the first N elements of the vector, the second row the next N elements, and so on.
- With images formed in this manner, we can express a broad range of linear processes applied to an image by using the notation

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}$$

Image Transforms

- All the image processing approaches discussed thus far operate directly on the pixels of the input image; that is, they work directly in the *spatial domain.*
- In some cases, image processing tasks are best formulated by transforming the input images, carrying the specified task in a *transform domain,* and applying the inverse transform to return to the spatial domain.

- Forward Transform

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) r(x, y, u, v)$$

- Inverse Transform

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) s(x, y, u, v)$$

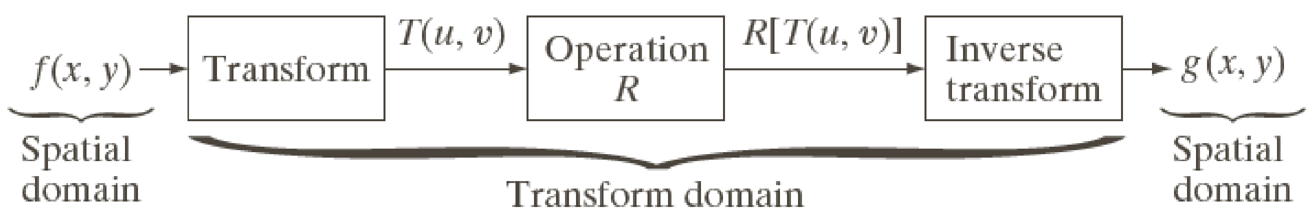


FIGURE 2.39 General approach for operating in the linear transform domain.

Example

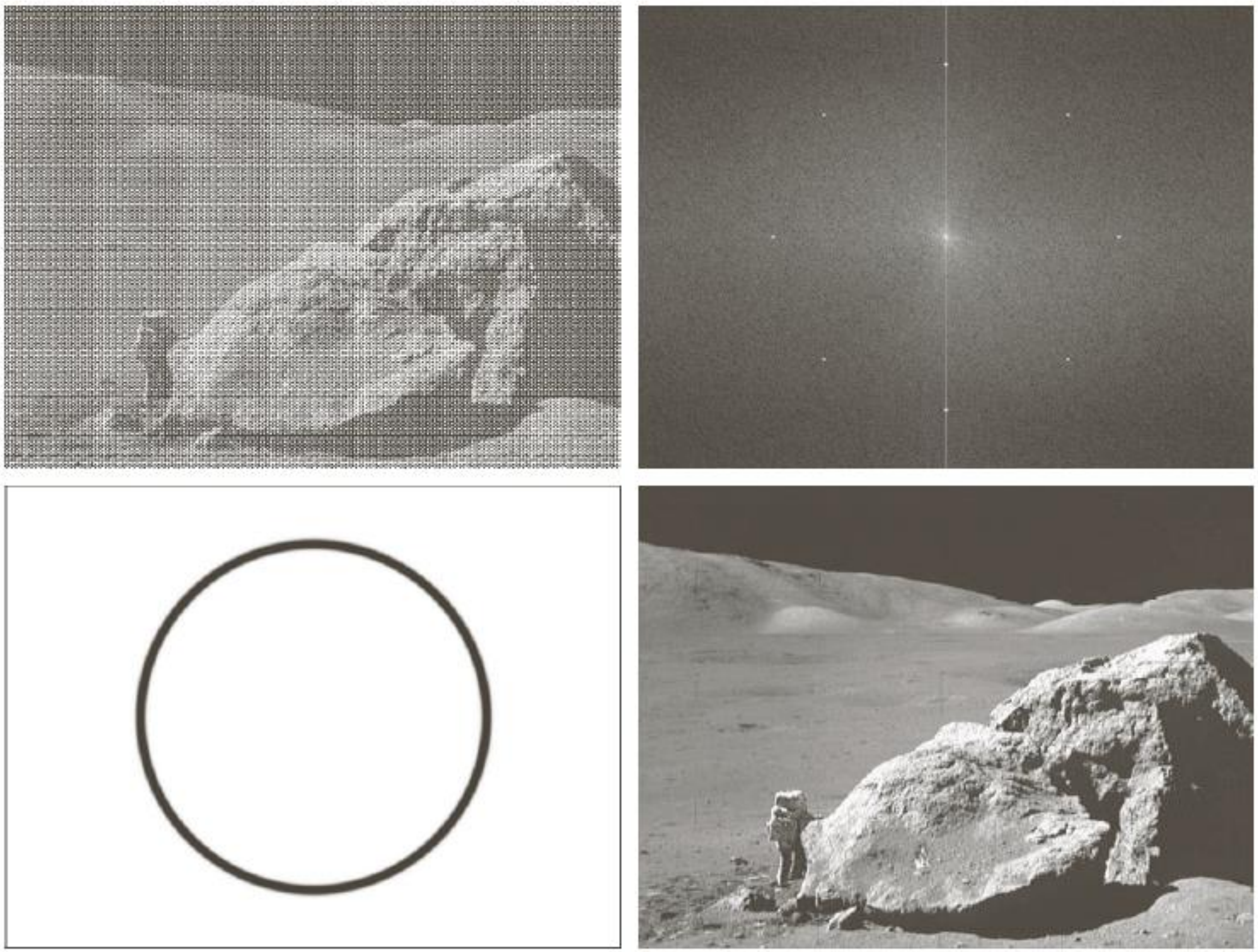


FIGURE 2.40
(a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy of NASA.)

Transform Kernels

- The forward transformation kernel is said to be separable if

$$r(x, y, u, v) = r_1(x, u) r_2(y, v)$$

- The kernel is said to be symmetric if

$$r(x, y, u, v) = r_1(x, u) r_1(y, v)$$

- The 2-D Fourier Transform has the following kernels

$$r(x, y, u, v) = e^{-j2\pi(ux/M + vy/N)}$$

$$s(x, y, u, v) = \frac{1}{MN} e^{j2\pi(ux/M + vy/N)}$$

- Discrete Fourier Transform pair:

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) e^{j2\pi(ux/M + vy/N)}$$

Matrix Form

- When the forward and inverse kernels of a transform pair are separable and symmetric, and $f(x,y)$ is a square image of size $M \times M$, then the forward transform

$$T(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) r(x,y,u,v)$$

can be expressed in matrix form: $\mathbf{T} = \mathbf{AFA}$

- To obtain the inverse transform, $\mathbf{BTB} = \mathbf{BAFAB}$

If $\mathbf{B} = \mathbf{A}^{-1}$

$\mathbf{F} = \mathbf{BTB}$: perfect reconstruction

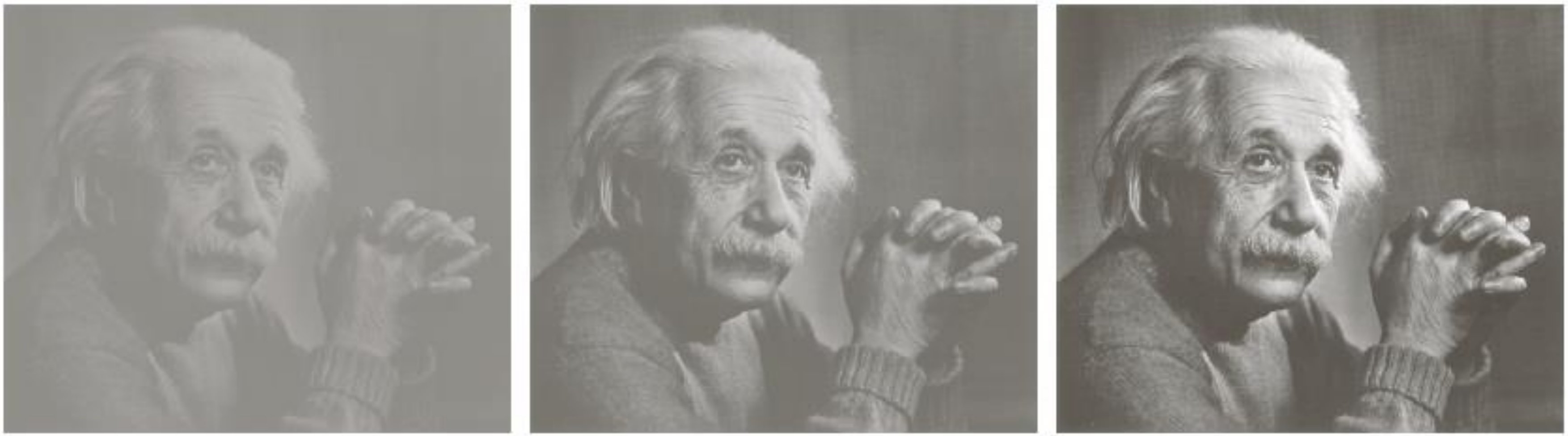
Otherwise, we have an approximation

$$\hat{\mathbf{F}} = \mathbf{BAFAB}$$

Probabilistic Methods

- Probability finds its way into image processing work in a number of ways.
- The simplest is when we treat intensity values as random quantities. For example, let z_i, $i = 0,1,2, \dots, L-1$, denote the values of all possible intensities in an $M \times N$ digital image. The probability of intensity level z_k in a given image is estimated as $p(z_k) = \frac{n_k}{MN}$.
- The mean (average) intensity is given by $$m = \sum_{k=0}^{L-1} z_k \, p(z_k)$$
- The variance: $$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2 \, p(z_k)$$
- The nth moment: $$\mu_n(z) = \sum_{k=0}^{L-1} (z_k - m)^n \, p(z_k)$$

STD as Measure of Intensity Contrast



14.3   31.6   49.2

a b c

FIGURE 2.41
Images exhibiting (a) low contrast, (b) medium contrast, and (c) high contrast.

Matlab Image Processing Toolbox

- Image Processing Toolbox User's Guide
  - Basic Image Import, Processing, and Export
  - Basic Image Enhancement and Analysis Techniques
  - Introduction
  - Reading and Writing Image Data
  - Displaying and Exploring Images
  - Building GUIs with Modular Tools
  - Geometric Transformations
  - Image Registration
  - Linear Filters
  - Transforms
  - Morphological Operations
  - Analyzing and Enhancing Images (Edge Detection, Image Segmentation, …)
  - Image Deblurring
  - Color
  - Block Processing, Code Generation, GPU Computing, …

Example Functions

- I = imread('coins.png');
- whos I
- imshow(I)
- Imtool(I)
- imwrite (I2, 'pout2.png');
- Imfinfo ('pout2.png');
- K = imfinfo('yellowlily.jpg');

  image_bytes = K.Width * K.Height * K.BitDepth / 8

  Compressed_bytes = K.FileSize

  Compression_ratio = image_bytes / Compressed_bytes